AMIGA DOS
ISSN 0937-2717
DMV-Verlag
DM 6,50 öS 52 sfr 6,50 Lit 6000 Ptas. 520
3/92

AMIGA DOS

3/92

DIE GANZE POWER

SPIELE-SHOW CES
Neuheiten aus Las Vegas

Computer-Lexikon
Amiga von A bis Z

Tips & Tricks
Wissen kurz und knackig

GRAFIK-KURS
DPaint IV sofort im Griff

AUDIOMASTER IV
Digi-Sounds total

PD-POWER
Masse mit Klasse

MUSIK-WERKSTATT
Tonstudio für zu Hause



PROGRAMMIERUNG
Welche Sprache wann, wie, wo?



Der Amiga ...

... das unbekannte Wesen? Das muß nicht sein! Sicher, ein wenig Geduld und eine kleine Portion Ehrgeiz gehören schon dazu, wenn Sie sich mit Ihrem Rechner etwas intensiver beschäftigen wollen. Wenn Sie dann auch noch Ihre Anstrengungen von Erfolg gekrönt sehen wollen, so werden Sie auf Hilfestellung nicht ganz verzichten können. Und die können Sie bekommen, denn Sie haben ja uns – und die AMIGA DOS.

Wir wollen Ihnen zeigen, wieviel Spaß man mit dem Amiga haben kann, wenn man versucht, ihn ein wenig zu "verstehen". Daß dies gar nicht so schwierig ist, beweisen wir Ihnen auf den folgenden Seiten. Wir glauben, daß wir wieder einmal einen interessanten Mix zusammengestellt haben, der für jeden etwas bereithält.

Titelthema der neuen AMIGA DOS: Programmieren. Wir liefern Ihnen eine Übersicht der wichtigsten Sprachen, zeigen ihre Einsatzmöglichkeiten auf und unternehmen einen kurzen Streifzug durch ihre Entstehungsgeschichte.

Tips & Tricks – noch nie so wertvoll wie heute! Wir haben jede Menge "Quickies" auf Lager, die Ihnen helfen, mit wenig Aufwand verblüffende Ergebnisse zu erzielen.

Sie haben Cola und Kartoffel-Chips, wir liefern den Rest: Mit Listings, Beispielen und Anleitungen machen wir den Programmiererseelen unter Ihnen das Leben leichter.

Was wäre die AMIGA DOS ohne Kurse? Gleich vier haben wir für Sie parat: Shell – die Perle Ihres Amiga; DPaint IV – simpler als Malen nach Zahlen; ARexx – Text ruft Grafik; und last but not least: Betriebssystem OS 2.0 – wie kommt System in den Betrieb?

Ob Violinkonzert oder Heavy Metal, der Amiga hat's drauf. Und Sie auch – spätestens dann, wenn Sie unseren Musik-Workshop studiert haben. Lesen Sie alles über die bedeutendsten Sampler, ihre Vor- und Nachteile und natürlich darüber, wozu und wie man sie einsetzt. In Vegas war die Hölle los, wir waren trotzdem dort. Im Spieleteil erfahren Sie, welche Neuerscheinungen Sie im Frühjahr erwarten. Außerdem natürlich: Tests, Tips, News & Facts.

Groß in der Qualität, klein im Preis – PD wird immer wichtiger. Wir haben noch mehr Seiten für Sie reserviert und präsentieren Ihnen das Neueste vom Neuen, vom Spiel bis zum Hartgesottenen-Zubehör.

Wenn Sie nun Ihre neue AMIGA DOS durchblättern, werden Sie feststellen, daß wir einige Änderungen vorgenommen haben. So liegt jetzt beispielsweise der Spieleteil in der Heftmitte, Dachzeilen über den Überschriften informieren Sie auf den ersten Blick über den Inhalt jedes einzelnen Beitrags, und die verschiedenen Rubriken wurden noch schlüssiger auf das Heft verteilt. All dies zusammen bedeutet, daß die AMIGA DOS noch übersichtlicher geworden ist. Für Sie heißt das, daß die gesuchten Informationen immer schnell zu finden und einfach zu verarbeiten sind. Wir glauben, daß wir Ihnen mit der neuen AMIGA DOS ein noch besseres Heft anbieten können, und wünschen Ihnen damit viel Spaß.

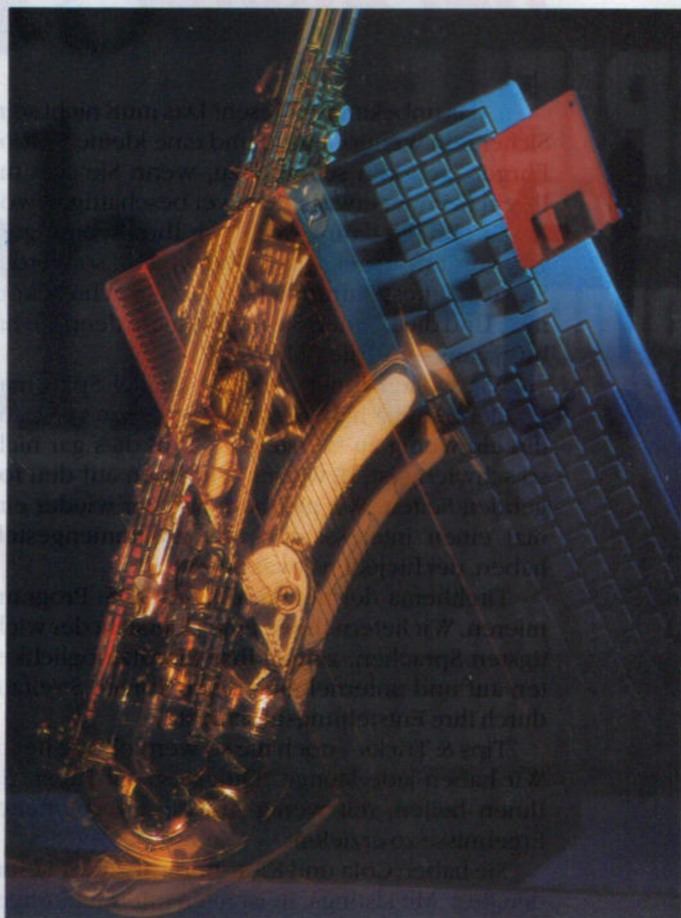
Tschüs für heute und bis zum nächstenmal

Bernd Zimmermann
Chefredakteur

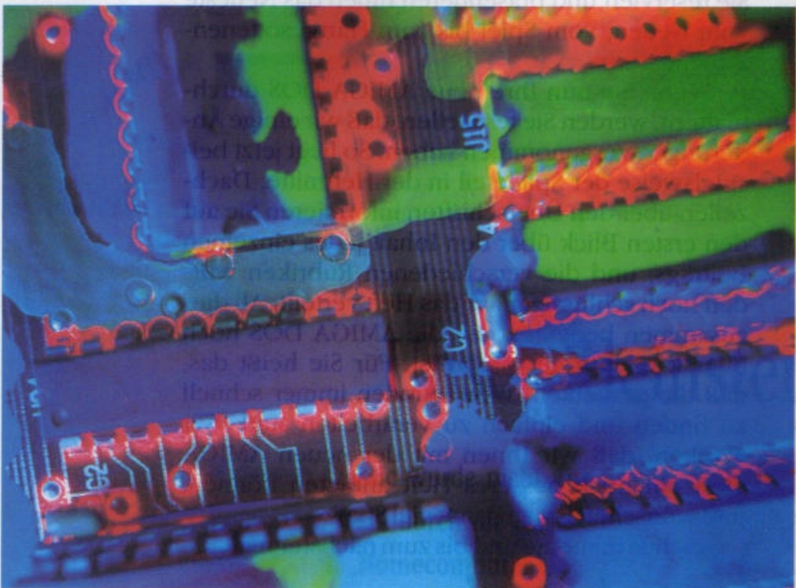
inh



Unser Titelthema beschäftigt sich mit Programmiersprachen, ihren Anwendungsmöglichkeiten und ihren Einsatzgebieten. Wie interessant das Thema "Programmieren" sein kann, zeigen wir Ihnen anhand einer Vorstellung der einzelnen Sprachen. Dabei zählen wir Vorzüge, aber auch Nachteile auf. Zu Wort kommen Programmierer, die sich mit "ihrer" Sprache bei Ihnen vorstellen wollen und Ihnen Positives und Negatives zu berichten haben.
Alle Informationen finden Sie ab Seite 10

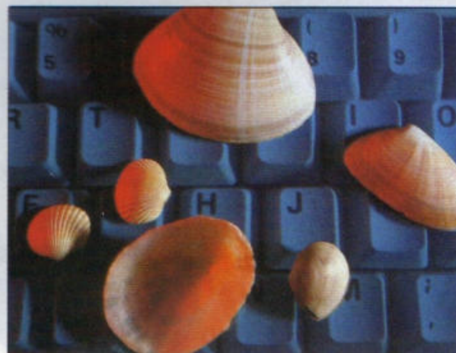


"Samplen" ist was Schönes – wenn man weiß, wie man einen Audio-Digitizer richtig bedient. In unserer Musik-Werkstatt geht es um Sampler, Sounds und tolle Hits. Bei uns sitzen Sie am Mikrofon und in der "ersten Reihe" – ab Seite 88



Im Kursteil haben wir unter anderem für Sie »Deluxe Paint IV« unter die Lupe genommen. Wir zeigen Ihnen, was mit diesem Malprogramm möglich ist. Das Zeichnen oder Erstellen von Bildern anhand der neuen Funktionen zeigen wir Ihnen durch viele Beispiele. Von der Installation bis zur Animation, von Pixeln bis zur Brush-Metamorphose werden alle Werkzeuge vorgestellt. Auch die Ausnutzung der grafischen Möglichkeiten im neuen HAM-Modus soll unser Thema sein, und zwar ab Seite 84

Mit der Shell läßt sich wesentlich mehr erreichen als unter der Workbench. Wie, das erfahren Sie ab Seite 102



aktuell

News

Titel

Test

Tips & Tricks

Programmierung

Spiele

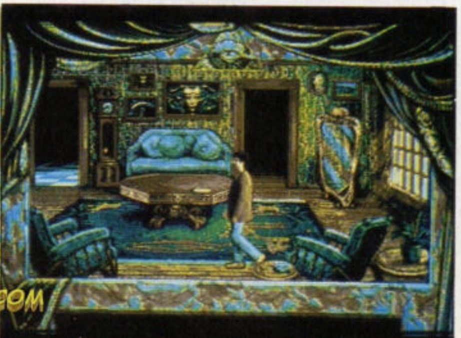
Kurs

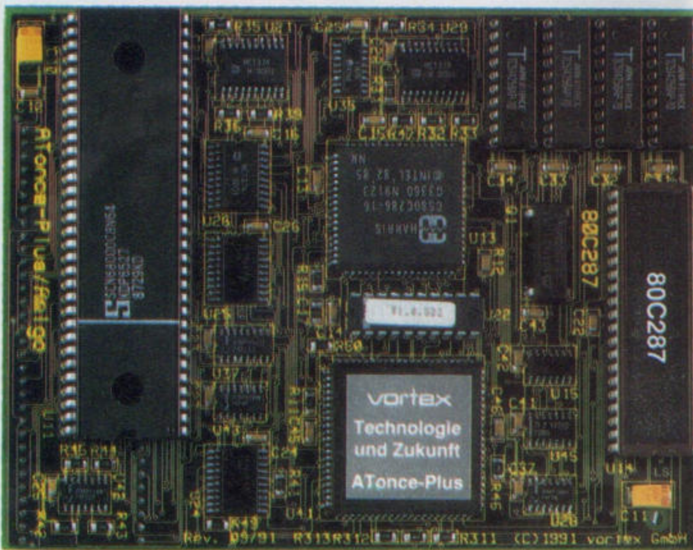
Workshop

Public Domain

Rubriken

Unter anderem im Spieleteil: »DarkSeed«, die Tür zu dunklen Dimensionen





Der AT-Emulator »ATonce Plus«

ATonce auch für Amiga 500 Plus

Die AT-Karte »ATonce« ist jetzt in einer neuen Version auch für dern Amiga 500 Plus erhältlich. »ATonce-Plus« verwaltet sowohl Festplatten als auch 3,5-Zoll- und 5,25-Zoll-Laufwerke mit 720 KByte oder 360 KByte. Es werden die Grafikmodi CGA (16 Farben), Hercules, Olivetti, Toshiba T3100, EGA-Monochrom (640x350) und VGA-Monochrom (640x480) emuliert. Der Amiga-500-Blitter wird voll eingebunden. »ATonce Plus« unterstützt Maus, Uhre, COM1 und COM2, LPT1 als parallele Schittstelle, das CMOS-RAM und Sound.

Info:
Vortex Computersysteme, Falterstr. 51-53. W-7101 Flein bei Heilbronn, Tel.:07131-59720

Gewinner AMIGA DOS 1/92

5 x beide Joysticks im Kombipack
Lars Wesche,
O-4090 Halle Neustadt
Jan Lange,
2102 Hamburg 93
Holger Kleinegräber,
4840 Rheda-Wiedenbrück
Jürgen Nüske,
7990 Friedrichshafen
Claudio Di Gaudio,
4100 Duisburg 11

10 x je ein Manix-Deck
Ronald Gindullis,
O-2500 Rostock 1
Musiol Wendelin,
5000 Köln 91
Jens Glock, O-5083 Erfurt
Christian Male,
A-9920 Sillian
Andreas Schnass,
4000 Düsseldorf 1
Harald Kauert,
2000 Hamburg 62
Max Larssen,
6101 Reichelsheim
Joachim Franz,
8500 Nürnberg 60
Patrick Mathae,
O-5912 Seebach
Udo Vieler, 5778 Meschede

10 x je ein Manix-Twin
Christian Zindel,
CH-7000 Chur
Matthias Kost,
O-7031 Leipzig
Wolfgang Reindl,
8079 Buxheim
Markus Kratz,
1000 Berlin 30
Marc-Oliver Wibbimek,
8552 Höchstadt
Michael Witte,
4512 Wallenhorst
Regina Link,
4000 Düsseldorf 1
Karin Makel,
5000 Köln 1
Elisabeth Lindow,
2000 Hamburg 55
Hermann Maier,
8200 Rosenheim

Computer-Meeting in Heilbronn

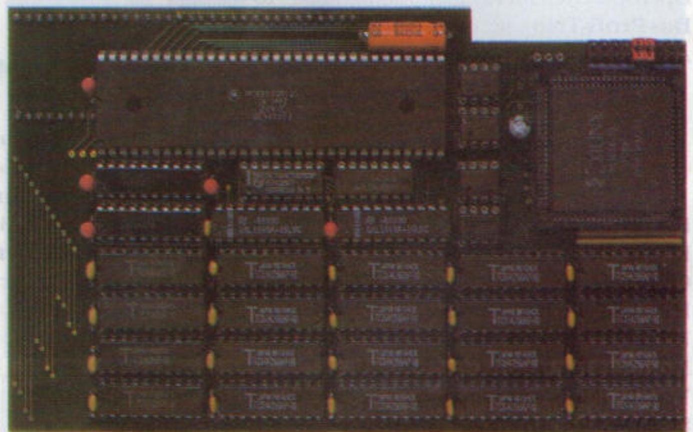
Der computer club 86 e.V. in Heilbronn veranstaltet wie jedes Jahr ein Computer-Meeting in Heilbronn. Das Meeting findet vom 28. bis zum 31. Mai 1992 von 10.00 Uhr bis 22.00 Uhr im Wilhelm-Waiblinger-Haus, Schützenstraße, in 7100 Heilbronn statt. Es sind Einsteigerkurse zu OS 2.0 und MS-DOS sowie ein Scanner-Workshop vorgesehen. Für den Samstag ist ein Technik-Flohmarkt geplant, an dem jeder nach vorheriger schriftlicher Anmeldung teilnehmen kann. Anmeldungen und weitere Informationen gibt es beim computer club 86 e.V..

Info:
computer club 86 e.V., Postfach 1418, W-7100 Heilbronn, Tel.: 07131-45168 (Herr Dietz) oder 07131-79646 (Herr Böhmer)

Caligari-Raytracer in Junior-Version

Das Raytracing- und Animationsprogramm »Caligari« hat eine Junior-Version bekommen. Das Programm nennt sich »Caligari2« und basiert auf der Broadcast-Version 2.0. Die Junior-Version benötigt mindestens 2 MByte RAM. Die Bildberechnung erfolgt in 24 Bit, die Farbausgabe im HAM-oder HAM-E-Modus. Ein Interface zum »DCTV« ist ebenfalls integriert. Caligari2-Dateien lassen sich in die "große" Broadcast-Version einlesen und weiterverarbeiten. Ein deutschsprachiges Handbuch und eine VHS-Videocassette sollen einen schnellen Einstieg in dieses Programm ermöglichen.

Info:
AEON Verlag und Studio, Fraunhofer Straße 51b, W-6450 Hanau 1, Tel.: 06181-23525


»Blizzard« sorgt für RAM und Geschwindigkeit

Speicher und Geschwindigkeit

Eine preiswerte Kombination von Speichererweiterung und Prozessorbeschleunigung bietet die Firma Advanced Systems & Software an. Die »Blizzard« genannte Karte bringt den Amiga-Modellen 500 und 2000 die doppelte Taktfrequenz (14,28 MHz) und bis zu 8 MByte RAM. Weitere 512 KByte RAM können zur Spiegelung des Betriebssystems benutzt werden. Der Einbau erfolgt lötfrei in den Prozessorsockel. Ein der Taktrate entsprechender Prozessor ist bereits auf der Karte montiert. Als Speicherbausteine werden 256 KBit x 4 oder 1 MBit x 4 mit 100 ns Zugriffszeit verwendet. Der Preis beträgt 329,- DM.

Info:
Advanced Systems & Software, Homburger Landstraße 412, W-6000 Frankfurt 50, Tel.: 069-5488130

Deutscher DICE-Distributor

Ab sofort gibt es für den bekannten Shareware-C-Compiler »DICE« von Matt Dillon einen deutschen Distributor. Die Registration kostet 85,- DM und beinhaltet ein freies Update.

Info:
Stefan Becker, Holsteinstraße 9, W-5100 Aachen, Tel.: 0241-505705

Scanner in Grau

Einen universellen Handscanner für IBM und Amiga bietet Reis-Ware an. Das »Scanking« genannte Gerät erreicht Auflösungen von 50 bis zu 800 dpi und zwei bis 64 Graustufen. Der 429,- DM teure »Scanking« wird mit Grafik- und Texterkennungssoftware ausgeliefert.

Info:
Reis-Ware, Postfach 36, W-5584 Bullay, Tel.: 06542-2086

Amiga 2000 Tower

Eine Tower-Version des Amiga 2000C bietet die Firma X-Pert an. Das Gehäuse wurde speziell für den Amiga 2000 entwickelt. Alle Anschlüsse sind rückseitig herausgeführt. Erhältlich ist der Tower in Beige oder Schwarz als Komplettgerät inklusive A2000C-Platine oder als Gehäusebausatz.

Info:
X-Pert Computer Services GmbH, Weiherwiese 27, W-6270 Idstein, Tel.: 06126-8809

A+L-Produkte mit Update-Abo

Die Schweizer Firma A+L AG, bekannt durch ihre Modula-2- und Oberon-Produkte, bietet ihren Kunden ein Update-Abonnement an. Für den doppelten Preis eines einfachen Updates bekommt der registrierte Anwender bis zu fünf Updates im Jahr automatisch zugesandt.

Info:
A+L AG, Däderitz 61, CH-2540 Grenchen, Tel.: 0041-65-520311



Die Hit-Mouse

Preisänderung

Wie uns die Firma Sunnyline Computer Products mitteilte, beträgt der Preis für die Hit-Mouse nicht, wie fälschlicherweise in AMIGA DOS 2/92 angegeben, 39,- DM, sondern 59,- DM.

Info:
Sunnyline Computer Products GmbH, Brönninghauser Straße 35d, W-4800 Bielefeld 16, Tel.: 0521-391131

Die Gewinner der AMIGA-DOS-Leserumfrage

Hier sind sie nun, die Gewinner aus AMIGA DOS 12/91 und 1/92. Die Gewinner erhalten die Preise per Post. Herzlichen Glückwunsch und viel Spaß!

RocGen Genlocks:
Steinke, Hanno
O-7031 Leipzig
Bez, Ingo
W-7122 Besigheim

ASM_ONE:
Frohms, Ralf
W-3180 Wolfsburg 1
Reebs, Heiko
W-8047 Karlsfeld
Leskau, H.-A.
W-2155 Jork
Freitag, Friedhelm
W-2400 Lübeck 1
Wegner, Gerhard
O-5210 Arnstadt
Ahr, Norbert
W-6000 Frankfurt 1
Hollmann, Axel
W-4520 Melle 1
Teegen, Uwe
W-3300 Braunschweig
Mohrmann, Wilfried
W-5600 Wuppertal 12
Nink, Bernd
W-6000 Frankfurt 50

Fraktal-Generator 3D:
Böhm, Robert
W-8000 München 80
Harder, Christian
W-2362 Wahlstedt
Chudalla, Alexander
W-7406 Mössingen
Echtenbruck, Dieter
W-4100 Duisburg 12
Waldbauer, Marcus
W-6660 Zweibrücken
Meyer, Bernd
W-6682 Ottweiler
Wietschovke, Thomas
W-4100 Duisburg
Rolle, Maik, O-8600 Bautzen
Nowak, Boris
1000 Berlin 47
Fischer, Eduard
W-2150 Buxtehude

Amiga-Spiel (für die Amiga-Mäuse):
Peter, Frank
W-3000 Hannover 91
Straub, Christian
W-5300 Bonn 02
Müller, Jürgen
W-6800 Mannheim 61
Schulze, Marc
W-4030 Ratingen
Janke, Timo
W-5860 Iserlohn
Stich, Kai
W-3150 Peine
Maier, Johann
W-8387 Roßbach
Peters, Uwe
W-2110 Buchholz
Paul, Günter
W-6452 Hainburg 1
Hög, Norbert
W-5650 Solingen

Jahres-Abo:
Limberg, Michael
W-4000 Düsseldorf
Wühr, Klaus
W-8000 München 80
Geissler, Bernd
W-7632 Friesenheim
Töpel, Brian
W-7800 Freiburg
Kell, Udo
O-7042 Leipzig
Hinz, Ulrich
O-2556 Gr. Freienholz
Rausweiler, Udo
W-6559 Odernheim
Müller, Claus
W-2050 Neu Börnsen
Lettner, Herbert
W-5905 Freudenberg
Freissler, Gerhard
W-7343 Kuchen

Scanner in 3D

Der »Profi-Line Scanner DS 3000« der Firma Macziejek Langer Computer ist durch seine spezielle Bauform in der Lage, flache und dreidimensionale Vorlagen abzutasten. Dabei werden 4096 Farben erkannt. Die Anwendungsbereiche für Scanner werden damit enorm erhöht: Katalogerstellung und Objektarchivierung sind nur wenige Beispiele. Die Datenübertragung erfolgt wahlweise über die parallele oder die serielle Schnittstelle. Aus diesem Grund ist die Installation schnell und denkbar einfach durchzuführen. So kann zum Beispiel eine Automodellsammlung oder die Veränderung einer Produktverpackung im Laufe der Zeit erfaßt und ausgewertet werden (zum Beispiel in Verbindung mit einem IFF-Archivierungsprogramm). Dreidimensionale Vorlagen können bis zu 337 mm breit, 299 mm tief und 423 mm hoch sein. Der Scanner wird mit einer speziell für den Amiga entwickelten Software und Anschlußkabeln geliefert. Die Schwarzweißversion kostet 1149,– DM, in Farbe ist das Gerät für 1898,– DM erhältlich.



Dreidimensionale Vorlagen verarbeitet der »Profi-Line Scanner«

Info:
Macziejek Langer Computer, Im Ring 29, W-4130 Moers 3, Tel.: 02841-42249

AMAS 2 – MIDI und Sampler

Komplett softwaregesteuert ist der neue Sampler »AMAS 2« von der Firma Microdeal. In diesem Gerät sind ein MIDI-Interface und ein 8-Bit-Stereo-Sampler integriert. Die Kontrolle des Interfaces erfolgt über ein in der Sample-Software integriertes MIDI-Panel. Die Samplingraten des Samplers reichen von 3 kHz bis 55,9 kHz. Die Lautstärke des Eingangssignals wird, ebenso wie diverse Audiofilter, ausschließlich über die mitgelieferte Software geregelt. Zahlreiche Effekte wie Echo, Reverb, Chorus und Phaser stehen zur Nachbearbeitung von Samples zur Verfügung. Der Editor, der die PAL-Auflösung ausnutzt, zeigt unter anderem auch eine dreidimensionale Ansicht (Fast Fourier Transformation) der Samples an.

Info:
Micro Limited, PO Box 68, St. Austell, Cornwall, England, PL25 4YB, Tel.: +44-726-68020

Imagine 2.0

Das Raytracing- und Animationsprogramm »Imagine« ist voraussichtlich bei Erscheinen dieser Ausgabe bereits in der Version 2.0 lieferbar. Laut Memphis Computer Products, dem deutschen Distributor dieses Programms, wird die Version 2.0 vorläufig mit einem 300 Seiten starken englischen Handbuch ausgeliefert; ein deutsches Handbuch ist in Arbeit. Im Lieferumfang des Pakets befinden sich neben den vier Programmdisks fünf Disketten mit Objekten (Fonts, komplette Szenen inklusive Brushes), Texturen, Brushes und Bumpmaps.


Ein Imagine-Ergebnis

Im Programm selbst wurde einiges geändert. Die Bedienungsoberfläche wurde dem Workbench-2.0-Look angepaßt. Der Umgang mit Texturen und Brushes wurde wesentlich vereinfacht. Bisher waren auf ebenen und reflektierenden Flächen die Pixel der globalen Brush zu erkennen. Dies wird nun durch Interpolation der einzelnen Pixel ausgeglichen. Auch eine Simulation von Nebel wurde integriert.

Dieser Nebel kann nicht nur als globale Option benutzt, sondern auch als Material für Objekte (zum Beispiel Wolken oder diffuse Lichtquellen) eingesetzt werden. Testberechnungen können jetzt direkt aus dem Editor heraus gestartet werden. Texte können direkt mit den Standard-Amiga-Fonts eingegeben werden, die sofort in 3D-Objekte umgewandelt werden. Punkte können ab jetzt, wie bereits von den Objekten bekannt, gruppiert und interaktiv editiert werden. Alle Requester für Farbeinstellungen besitzen ab dieser Version ein Anzeigefeld der entsprechenden Farbe. Objekte können nun nicht nur um Kugeln oder Zylinder gewickelt, sondern auch um frei definierbare Pfade. DCTV wird von »Imagine 2.0« direkt unterstützt; auch 24-Bit-Animationen können direkt abgespielt werden.

Benutzer der Versionen bis 1.1 sollen sich, so behauptet Memphis, ohne größere Handbuchstudien sofort im Programm zurechtfinden. Die Update-Preise zu älteren Versionen bis 1.1 sind gestaffelt, und auch Turbo-Silver-Anwender können ein Update zum Sonderpreis erhalten.

In Kürze ist "Das Imagine-Buch" erhältlich, das auf alle neuen Features der Version 2.0 eingeht.

Info:
Memphis Computer Products GmbH, Gartenstraße 11, W-6365 Rodheim v. d. Höhe, Tel.: 06007-7789

Tolle Angebote für AMIGA-Gourmets.

"Am Stück oder geschnitten?"

An alle CDTV-User

Neu

Das VECTOR-CDTV Maus-& JoyStick Interface

- Anschl. f. stand. Mäuse, Joysticks, Trackballs an CDTV
- Incl. Back-Panel für zwei Geräte

Top-Preis DM 79,–

Neu

Farb-Genlock

...incl. komfortabler TitelSoftware

- Super Effekte
- Ein-/Aus-/ Überblendregler
- FBAS m. Chinch-Buchsen
- Abgeschirmtes Metallgehäuse

Top-Preis DM 299,–

Neu

Das VECTOR-Track A2000

- Trackdisplay f. A2000 zum Selbsteinbau in Power-LED-Träger
- Zeigt alle vier Disk.-Laufwerke an
- Laufwerks-Nrummern einstellbar

Top-Preis DM 98,–

Der VECTOR-Maus-& JoyStick-Adapter

...automatisch!



A500 / 1000 / 3000	A2000 / 2500
DM 24,50	DM 29,–

Das VECTOR-RAM Board 512 KB A500i

Top-Preis DM 69,–

Das VECTOR-RAM Board 1 MB für A500 Plus

Top-Preis DM 148,–



Das VECTOR-RAM Board 2 MB A500i

- 2 MB RAM
- bestückt mit 4 MBit–Chips
- Lauffähig mit 512 KB und 1 MB Chip-Memory
- Incl. GARY–Adapter

Top-Preis DM 298,–

Sound-Digitizer der Spitzenklasse!

Der VECTOR-SOUND-Digitizer

- Mono & Stereo
- Eigener, superschneller A/D-Wandler je Kanal
- Spannungsversorg. "On Board"
- Metallgehäuse
- Lange Zuleitung
- CINCH-Eing.
- Eingangs-Signal je Kanal einstellbar

Besuchen Sie uns auf unserem Messe-Stand

Hobbytronic '92, Dortmund
25.3.–29.3.92

AMIGA '92, Berlin
2.4.–5.4.92



incl. komfortabler Software

Top-Preis DM 298,–

Das VECTOR-RAM Board Max 8 MB A500i



- Bis max 8 MB zusätzl.
- Bestückbar in 512 KB bzw. 2 MB Schritten, auch gemischt
- Echtes Autokonfig.
- Echtes FastRAM
- Hard- und Softwaremäßig abschaltbar
- CPU-Adapter
- Läuft zusammen mit Erweiterungen im RAM-Slot
- Bestückt mit 0 MB

Top-Preis ab DM 298,– (...bestückt mit 0 MB)

Wie's beliebt!

512 KB RAM-Satz DM 65,–

2 MB RAM-Satz DM 199,–

Das VECTOR-RAM Board Max 8 MB A2000i

- Echtes Autokonfig.
- Echtes FastRAM
- Hard- und Softwaremäßig abschaltbar
- Bis max 8 MB
- Bestückbar in 512 KB bzw. 2 MB Schritten, auch gemischt
- Verpolungssichere Steckkarte
- Vergoldete Kontakte
- Bestückt mit 0 MB



Top-Preis ab DM 179,– (...bestückt mit 0 MB)

Die VECTOR-DRIVEs

3 1/2" Laufw. AMIGA 500 intern	DM 198,–
3 1/2" Laufw. AMIGA 2000 intern	DM 129,–
3 1/2" Laufw. für alle AMIGAs extern	DM 149,–
5 1/4" Laufw. für alle AMIGAs extern	DM 199,–

Die VECTOR-Bremsen

AMIGA-Bremse intern f. alle AMIGAs	DM 39,50
AMIGA-Bremse f. A500 ext. m. LED	DM 39,50

Das VECTOR-Kick-ROM 2/3

(vorber. für Kick-Start 2.x)

- Umschaltplatine für 2 Orig.-ROMs — **DM 49,–**
- Umschaltplatine für 3 Orig.-ROMs — **DM 59,–**
- Kick ROM 2/3 m. ROM 1,3/2.0 — **DM 98,-/108,-**

Das VECTOR-Midi-Interface

...für alle AMIGAs. Mit 700% Kopplungsfaktor incl. exzellenter Sequenzer-Software **DM 199,–**

Der VECTOR-Turbo-Call

Der AMIGA als Anrufbeantworter!

- Beliebige Ansagetexte
- Speicherbare Tel.-Nr. — **DM 89,–**

Die VECTOR-Boot-Selectoren ELEKTRON / STANDARD

Der VECTOR-Boot-Selector ELEKTRON:

- Wahlweise booten von allen Laufwerken
- DFO: weiterverwendb./abschaltb. — **DM 49,–**

Der VECTOR-Boot-Selector STANDARD:

- Wahlweise booten von DFO: oder DF1: oder DF2: oder DF3: (b. Bestell. bitte angeb.) — **DM 14,50**

Der VECTOR-Multi-Port-Adapter

Für beide Ports gleichzeitig • Geeignet für Mäuse, Joysticks, Dongels, Btx usw. • Autom. u. man. Umschalt. • Dauerfeuer mit Impulsbreiten- und Pausenregelung **DM 148,–**

Externe Festplatte f. AMIGA 500/500 Plus

- AT-Bus Controller, 8 MB RAM-Option on Board, incl. Netzteil

... mit Quantum LPS 52 AT **DM 998,–**

... mit Quantum LPS 105 AT **DM 1.398,–**

Die VECTOR Filecard 52 MB

- Controller mit Quantum LPS 52 S; komplett installiert

DM 898,–

Die VECTOR Filecard 105 MB

- Controller mit Quantum LPS 105 S; komplett installiert

DM 1.298,–

Sofort! Persönliche Bestellannahme 0221/311606

PreisSturz!



OPTO-Maus	MegaMaus	CRYSTAL-Trackball	Trackball
Top-Preis DM 119,–	Top-Preis DM 59,–	Top-Preis DM 109,–	Top-Preis DM 99,–
Neueste Technik! Volloptische Maus (ohne Kugel), 300 dpi	Optomechanischer Dechiffrierer, 280 dpi	Optische Kontrolle der Tasten-Funktionen durch farbiges Aufleuchten (rot/grün) für AMIGA	Der Rrobuste für alle AMIGAs

Ihre Partner, unsere Stützpunkthändler ...

Nachnahme-Versand innerhalb Deutschland: Per Post DM 10,–; per UPS DM 20,–; ins Ausland DM 30,– ; Großgeräte nach Gewicht.

VECTOR–Produkte erhalten Sie auch bei:

Babe EDV · 5220 Waldbröl · 02291/5036 • Bernd Neumann, Computer-Center · 2953 Rhauderfehn · 04952/8520 • BIT Sommer & Diekmann · 4040 Neuss · 02101/275751 • Fischer Hard- und Software · 3000 Hannover · 0511/575087 • Hard 'n Soft GmbH · 4130 Moers · 02841/170150 • W&L Computer · 1000 Berlin · 030/6227371 • Büro & Technik · 0-2300 Stralsund · 0037/821/293345 • Heschko Computer · 0-4400 Bitterfeld · 0037/441/42116



F. Hansmann & Th. Küpper GbR
Bonner Str. 37 · 5000 Köln 1
Tel. Technik: 0221/3318157
Fax 0221/321166 · Btx *HK#
Mo–Fr 10⁰⁰–13³⁰, 14³⁰–18³⁰
Sa 10⁰⁰–14⁰⁰

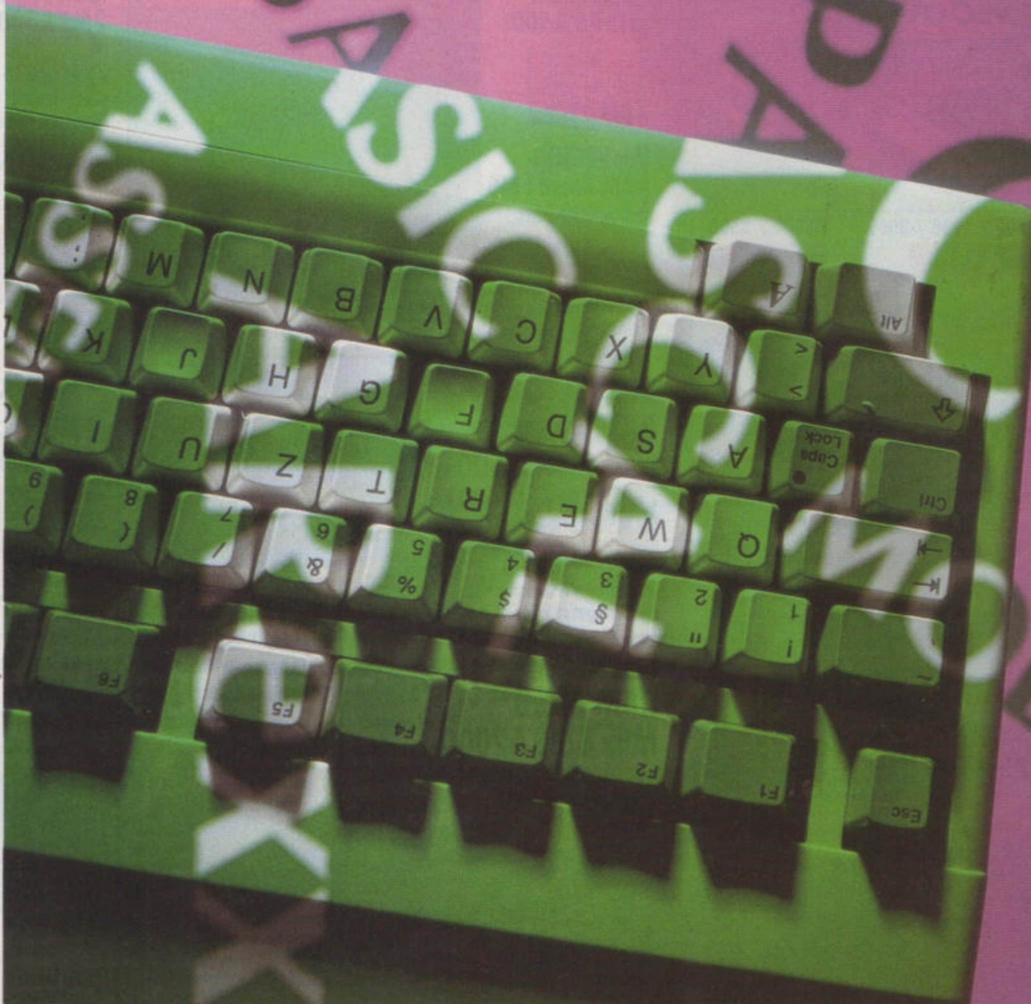
Autorisierter Commodore-Fachhändler
Commodore Commercial Developer



Bestellungen über Telefon: **0221 / 31 16 06**

Fordern Sie unseren Sortiments-Prospekt an!

Hinweis: Alle unsere externen Geräte haben keine ZZF-Zulassung, wenn nicht gesondert angegeben. Ein Betrieb im Bereich der Deutschen Bundespost ist verboten und strafbar.



Programmiersprachen

Programmierer haben's schwer

Wer ist schuld, wenn die Textverarbeitung genau in dem Augenblick abstürzt, wenn derjenige, der davorsitzt, auf die Funktion "Save as..." schaltet? Natürlich die Programmierer! Doch wer denkt schon daran, wieviel Mühe das Erstellen des Programms gemacht hat? Wer? Na klar – der, der vorher kräftig geschimpft hat – wenn er seine ersten Gehversuche macht.

Die Wünsche sind vielfältig: Malprogramm, Sound-System, Textverarbeitung, DTP-Programm, Action-Spiel oder Adventure – die Umsetzung endet leider meist irgendwo in der Schublade. Wer gerade seinen neu erworbenenen Amiga auspackt und als Einsteiger oder Umsteiger jetzt in Eigenregie Programme erstellen will, erlebt erst einmal eine herbe Enttäuschung: AmigaBASIC, bisher enthalten auf der »Extras-Disk«, gibt's nicht mehr. Nur die glücklichen Besitzer eines A3000 oder A500 plus bekommen einen "Ersatz": ARexx.

"BASIC", "ARexx" – Begriffe wie aus einem fernen Land. Aber es kommt noch schlimmer, denn trifft man Leute, die den Amiga schon besser kennen als ihre Westentasche, so geht es gleich los: "BASIC? Doch nur was für Kinder!" "Assembler? Schon mal in einem Rennwagen ohne Bremse ge-

sessen?" – Kurzum, es dauert nur wenige Gespräche lang, und man weiß nicht mehr, woran man ist.
Möglichkeiten, den Amiga zu programmieren, gibt es viele; genauer gesagt sind es die Programmiersprachen, die den verschiedenen Anforderungen entsprechen können. Leider gibt es da immer noch ein Problem: Als Einsteiger versteht man immerzu "Bahnhof", selbst dann, wenn die erste Zeile eines BASIC-Programms nur aus einem Kommentar besteht. Programmiersprachen müssen sinnvoll eingesetzt werden, wenn sie Sinnvolles erzeugen sollen.

Es schwirrt einem der Kopf

Wer seinen Computer nur für das Laden des neuesten Shoot'em-ups benötigt (manche nennen dieses Spielgenre auch abfällig "Baller-Ware", gemeint sind Schießspiele aller Art), weiß manchmal gar nicht, welche Möglichkeiten er außen vor läßt. Ob Beruf oder Schule, Studium oder Hobby, fast immer kommt es vor, daß bestimmte Vorgänge immer wiederkehren.
Für meine beiden liebsten Hobbys, Modelleisenbahnen und Elektronik (wobei beide erst nach meinem Amiga kommen), ergeben sich immer wieder Situationen, in denen sich der Rechner als Schalt- oder Verwaltungsstelle bewährt.
Wer selbst Hobby-Elektroniker ist, der weiß, wieviel Kleinmaterial sich ansammeln kann. Da werden beim Versender A zum Beispiel 200 Leuchtdioden bestellt (die waren so billig, und da man ja ohnehin zwei Stück für die Schaltung braucht ...), beim Laden B an der Ecke kauft man immer gern die Sortimentskisten mit den 2000 Widerständen mit verschiedenen Werten.

Programmieren heißt logisch denken

Und zu Hause fliegen noch irgendwie die Transistoren und integrierten Schaltungen herum, die den Ersatzteilstatus für den Fernseher haben – kurzum, ein Programm muß her, das die Elektronikbauteile verwaltet und automatisch Bestellscheine erstellt, die auf dem Drucker ausgegeben werden. Da solche Programme in der Wirklichkeit kaum zu bekommen sind (selbst die PD-, FD- oder Shareware-Disketten bieten für solche Sonderfälle kaum eine Lösung an), bleibt einem nichts anderes übrig, als sich selbst an die Arbeit zu machen. Bei Modelleisenbahnen (als Beispiel) sieht es genauso aus.
Obwohl viele Hersteller darangehen, ihre Modelle für den Empfang von digitalen Daten auszurüsten und dementsprechend auch schon Interfaces (Verbindungen) hergestellt haben, braucht wohl jeder Modellbahn-Fan oder -Club ein Steuerprogramm, das nach den Gegebenheiten funktionieren muß. Und Steuerungsaufgaben sind teilweise auch noch Domäne der direkten Prozessorprogrammierung, also Assembler.
Obwohl beide Beispiele ziemlich speziell sind, zeigen sie doch eines auf: Beide benötigen eine spezielle Programmierungsart und somit auch bestimmte Eigenarten einer Programmiersprache.

Jede Sprache hat Licht und Schatten

Und damit sind wir beim Kern des Ganzen. In den nachfolgenden Artikeln geben wir Ihnen einen Einblick in die Programmierung sowie in die Möglichkeiten der jeweiligen Sprache. Über eines müssen Sie sich allerdings im klaren sein: Auch Autoren sind mitunter Programmierer, in die nachfolgenden Beiträge werden also sicher auch die persönlichen Vorlieben des Verfassers einfließen.
Für welche Sprache Sie sich letztendlich entscheiden, hängt von Ihnen ab.
Was wir Ihnen aber geben, sind Informationen über die Einsatzgebiete und Stärken und Schwächen der einzelnen Programmiersprachen. Gleichzeitig zeigen wir Ihnen, wie Sie selbst kleine Versuche wagen können. Und ein wenig Geschichte der Programmiersprachen ist auch dabei.
Schließlich möchten wir noch gern von Ihnen wissen, ob Sie schon Erfahrung mit Programmiersprachen haben oder Neuland auf diesem Gebiet betreten. Zu diesem Zweck schauen Sie sich doch die kleine Umfrage anschließend an. Mit ihr geben Sie uns Aufschluß darüber, was Ihre Vorlieben beim Programmieren sind und worüber Sie mehr wissen möchten, und sich selbst, was Ihnen zum Thema Amiga und Programmieren so vorschwebt.
Also, auf in den Kampf mit Befehlen, Strukturen und Adressierungen. ❑

(jb)

Programmiersprachen – Ihr Hobby?

Uns Redakteure (die auch gerne programmieren) interessiert, welche Erfahrungen Sie, lieber Leser, liebe Leserin, schon mit Programmiersprachen hatten, ob Sie eine Sprache anwenden oder vorhaben, in die Programmierung einzusteigen. Aus diesem Grund haben wir eine kleine Umfrage eingefügt, die Sie ausfüllen und an uns schicken sollten. Am besten geht dies, wenn Sie diese Seite kopieren, dann brauchen Sie das Heft nicht zu zerschnippeln.
Da wir denken, daß unsere Leser fleißig mitmachen, verlosen wir unter allen Einsendern drei ASM-One-Assembler. Also, mitmachen lohnt sich in doppelter Hinsicht: 1.) Sie bestimmen, was wir für Sie in Zukunft bevorzugt bringen werden. 2.) Sie haben die Chance, bald aktiver Programmierer zu sein.
Einsendeschluß für die Umfrage ist der 15.03.1992, der Rechtsweg ist ausgeschlossen, und Mitarbeiter des DMV sowie deren Angehörige dürfen nicht mitmachen (die können 1.) sowieso den Amiga programmieren, und 2.) wenn 1.) nicht zutrifft, dann besitzen sie einen MS-DOS-Computer).

Und hier sind die Fragen:

1.) Welchen Amiga besitzen Sie:
○ Amiga 500 ○ Amiga 500 Plus
○ Amiga 2000/2500 ○ Amiga 3000

2.) Wie schätzen Sie sich selbst beim Programmieren ein?
○ Noch nicht probiert ○ Anfänger ○ Fortgeschrittener
○ Halbprofi ○ Profi

3.) Welche Programmiersprache kennen Sie/nutzen Sie/wollen Sie lernen?
Sprache 1: ______
○ kenne ich ○ programmiere ich ○ will ich lernen
Sprache 2: ______
○ kenne ich ○ programmiere ich ○ will ich lernen
Sprache 3: ______
○ kenne ich ○ programmiere ich ○ will ich lernen

4.) Über welche Sprache würden sie am liebsten mehr/gleichviel/weniger/gar nichts lesen?

	mehr	gleichviel	weniger	gar nichts
○ AmigaBASIC	○	○	○	○
○ AMOS-BASIC	○	○	○	○
○ Blitz-BASIC	○	○	○	○
○ PCQ-Pascal	○	○	○	○
○ Kick-Pascal	○	○	○	○
○ Modula-2	○	○	○	○
○ Oberon	○	○	○	○
○ C	○	○	○	○
○ Assembler	○	○	○	○

5.) Wie informieren Sie sich über Ihre Programmiersprache?
○ programmiere nicht ○ über Bücher
○ über Magazine ○ nur durch das Handbuch

6.) Jetzt nur noch Ihren Namen und Ihre Adresse:
– Name: ______
– Vorname: ______
– Straße: ______
– PLZ/Ort: ______
– Telefon: ______ / Alter: ______

Wie gesagt, kopieren Sie die Seite, schneiden Sie die Umfrage aus, und schicken Sie sie an folgende Adresse:
DMV-Verlag, Redaktion AMIGA DOS, Kennwort "Programmieren", Postfach 250, 3440 Eschwege.



Programmiersprachen

Das kybernetische Babylon

Man stellt sich das so leicht vor: Irgendwann einmal gab's Computer. Nur – programmiert werden mußten diese Dinosaurier der Technik ja auch. Wie – das sagt Ihnen unser kleiner Rückblick.

Irgendwann in ferner Vergangenheit (genauer gesagt Mitte der vierziger Jahre) begann die Entwicklung erster elektronischer Rechenmaschinen – der Computer war geboren.
Heutige Computer haben mit dieser ersten Generation von Rechenknechten, die kaum die Leistung eines modernen Werbegeschenk-Solarrechners hatten, scheinbar wenig gemein. Trotzdem finden sich dort bereits wesentliche Elemente, die auch heute noch jeder Programmiererin und jedem Programmierer bekannt sein dürften – zum Beispiel die binäre Logik oder eben das **Programm**.
Was ist nun ein Programm? Eigentlich nichts weiter als ein Ablaufplan mit Anweisungen. Ein Computer ist ein Gerät, um solche Programme abzuarbeiten. Er führt also die im Programmtext gegebenen Anweisungen aus.
Den ersten Rechnern mußten Programme noch über Lochkarten oder riesige Schalttafeln mitgeteilt werden – Arbeitsspeicher, mit dem heute jeder Heimcomputer reichlich gesegnet ist, gab es einfach noch nicht.
Wer sich in diesem Zusammenhang gefragt hat, warum Computer in "Raumschiff Enterprise" oder ähnlichen Filmen einen solchen Lärm machen: Tausende von Relais in den ersten Rechenmaschinen sorgten für eine derartige akustische Untermalung. Heutzutage ist das natürlich Unsinn; von einem Amiga in Betrieb hört man nur das sanfte Säuseln des Lüfters – allenfalls noch die mehr oder weniger dezent singende Festplatte.
Die theoretischen Grundlagen dieses Prinzips sind übrigens einige Jährchen älter – sie stammen unter anderem von der englischen Mathematikerin Ada Lovelace. Das Pentagon, Sitz des amerikanischen Verteidigungsministeriums, "ehrte" die Lady, indem sie eine für militärische Zwecke entwickelte Programmiersprache nach ihr benannte.

Plauderstündchen der Maschinen

Verstehen kann ein Computer nur Anweisungen in seiner **Maschinensprache**. Doch, welcher Mensch schreibt gerne Programme nach folgender Methode

```
0100001010111001
0000000000000000
0000000000000100
```

Für Bitgucker: Dies ist ein kurzer Virenkiller in der Maschinensprache der Amiga-CPU. Wie Sie sehen, wird auf diese Art und Weise kaum je ein sinnvolles Programm zusammenkommen. Man ging nun einen Schritt weiter und versah jeden Befehl der Maschinensprache mit einer Abkürzung, "Mnemonic" genannt. Meist sind diese an die (englischen) Funktionsbeschreibungen der Befehle angelehnt; im obigen Beispiel sähe das so aus:

```
CLR.L 4
```

Dies bedeutet: Lösche (englisch: CLeaR) ein Langwort (.L) ab der Adresse 4 – wobei "Langwort" die Bezeichnung für einen vier Byte langen Wert ist.
Das ist doch schon wesentlich lesbarer, oder? Diese Form der Notierung der Maschinensprache nennt man oft **Assembler-Sprache** oder nur kurz **Assembler**. "Der Assembler" wiederum ist ein Programm, das diese Befehle in das Maschinenspracheformat des Rechners umwandelt.
Die Maschinensprache ist aber auch in ihrer lesbaren Form zwangsläufig sehr primitiv und daher für komplexe Anwendungen eigentlich kaum geeignet.

Komfort-Assembler

Die Einschränkung "kaum" ist berechtigt, weil zum einen die Maschinensprache der MC680x0-Prozessorfamilie eigentlich sehr komfortabel und umfangreich ist, zum anderen gute Assembler mit Makro- und ähnlichen Funktionen die Programmierung sehr vereinfachen. Dafür hat die Maschinensprache einen unbestreitbaren Vorteil: Sie ist für jedes Problem – bei richtiger Anwendung – die mit Abstand schnellste und/oder kürzeste Lösung. Denn bei der Assembler-Programmierung wird ja automatisch die für jedes Problem optimale Lösung gesucht.
Zurück in die Vergangenheit. Das mit der Maschinensprache nicht allzugut zu programmieren war, merkten die Wissenschaftler bei komplexeren Aufgaben schnell. In der Konsequenz wurden dann Programmiersprachen entwickelt, die sich nicht mehr direkt an den Bedürfnissen der

RISC, CISC und Co.

Bei der Entwicklung der Prozessoren ging man verschiedene Wege. Zuerst mußten die Prozessoren intern sehr einfach aufgebaut sein, daher waren die Maschinensprachen sehr kryptisch.
Mitte der Siebziger Jahre wurden dann erstmals Prozessoren entwickelt, bei denen zuerst ein Konzept für eine "menschenfreundliche" Maschinensprache entwickelt und dann der Prozessor gewissermaßen "drumherumgebaut" wurde. Das Ergebnis dieser Überlegungen war zum Beispiel die im Amiga eingebaute MC68040-Familie.
Andere Firmen erweiterten einfach den Befehlssatz immer mehr, mit dem Ergebnis Inkompatibilitäten der verschiedenen Betriebsmodi, "illegalen" Befehlen und so weiter. Beispiele dafür: Z-80 (ZX-81, ZX-Spectrum) oder die Intel-Prozessoren (IBM-Kompatible).
Solche Prozessoren nennt man "CISC"-CPUs ("Complex Instruction Set", zu deutsch etwa "kompletter Befehlssatz").
In letzter Zeit machen sogenannte "RISC"-Prozessoren von sich reden: "Reduced Instruction Code", also "Reduzierter Befehlssatz". Diese Prozessoren haben nur sehr wenige Befehle, die dafür sehr schnell ausgeführt werden. Zwar sind sie "von Hand" relativ schwer zu programmieren, dafür können Compiler sehr schnelle Programme erzeugen.

Maschine orientierten, sondern am zu lösenden Problem. Für mathematische Aufgaben zum Beispiel entstanden Programmiersprachen wie zum Beispiel "Fortran". Problemorientierte Sprachen nennt man in der Informatik "Hochsprachen".
Da der Computer aber weiterhin nur Maschinensprache verstand, mußte ein Bindeglied her, daß die Anweisungen der Hochsprache in Maschinensprache übersetzte. Dafür gibt es **Compiler** und **Interpreter**. Compiler übersetzen ein Hochsprachenprogramm einmal in ein komplettes ablauffähiges Maschinenspracheprogramm. Interpreter dagegen arbeiten das Hochsprachenprogramm schrittweise ab und führen für jeden Befehl eine festgelegte Maschinensprache-Routine aus.
Compiler brauchen natürlich für die Übersetzung eine gewisse Zeit, dafür ist das compilierte Maschinenprogramm recht schnell. Ein Interpreter dagegen führt das Programm sofort aus, allerdings wesentlich langsamer. Zum Entwickeln und Testen eines Programms ist ein Interpreter besser geeignet, da Fehler schnell im Hochsprachenprogramm erkennbar und korrigierbar sind und das Programm so schrittweise bequem getestet und erweitert werden kann. Compiler wiederum erzeugen wesentlich schnellere Programme.
Eine besondere Art von Compiler sind die sogenannten P-Code-Compiler. Diese erzeugen aus dem Hochsprachentext ein Programm in einer simplen, aber schnelleren "Zwischensprache" (P-Code), die beim Ablauf dann von einem Interpreter-Modul ausgeführt wird.
Eine Zeitlang waren auch Prozessoren im Gespräch, die direkt eine Hochsprache verstehen konnten (zum Beispiel ein Forth-Prozessor). Von diesem Entwicklungspfad ist man aber wieder abgekommen und hat sich statt dessen Gedanken über die Weiterentwicklung der Maschinensprachen gemacht.

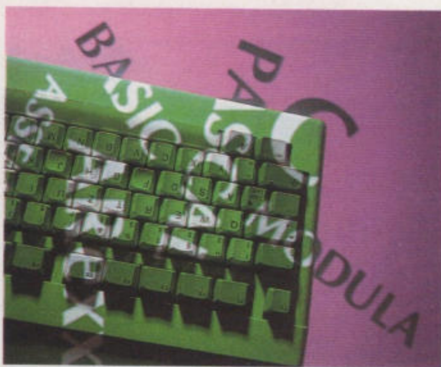
Wichtige Begriffe

In diesem und den nachfolgenden Texten tauchen Begriffe auf, die wir an dieser Stelle erklären wollen.
Assembler: Notationsform der -> Maschinensprache. Auch: Programm zum Übersetzen von Quellcode in Maschinensprache
Code: Oft Bezeichnung für ein ausführbares Programm in -> Maschinensprache
Compiler: Programm, das ein Hochsprachenprogramm in ein ablauffähiges Maschinenprogramm übersetzt. Siehe -> Interpreter
Debugger: Hilfsprogramm zur Fehlersuche in compilierten Programmen
Hochsprachen: Sprachen, die sich von der Konzeption her nicht am Befehlssatz der Maschine orientieren, sondern am zu lösenden Problem
Interpreter: Programm, das ein Hochsprachenprogramm schrittweise übersetzt und dann ausführt. Im Gegensatz dazu siehe -> Compiler
Linker: Programm zum Verbinden mehrerer Programmodule zu einem ablauffähigen Programm
Maschinennah: Als maschinennahe Sprachen bezeichnet man -> Hochsprachen, die trotz Problemorientierung Zugriff auf prozessornahe Elemente bieten (zum Beispiel C)
Maschinensprache: Sprache des Prozessors. Anzutreffen meistens in notierter Form (-> Assembler). Im Gegensatz dazu -> Hochsprachen
Optimizer: Hilfsprogramm zum Optimieren des von einem -> Compiler erzeugten Codes. Das geschieht durch Optimierung bestimmter Maschinensprachekonstrukte (sogenannte Peephole-Optimierung) oder Untersuchung und eventuell Umstellung des Programmflusses (globale Optimierung)
Profiler: Hilfsprogramm zur Untersuchung, welche Teile eines Programms die meiste Rechenzeit benötigen. Wird für Optimierungszwecke eingesetzt
Quelltext/Quellcode: Programmtext in einer Hochsprache vor der Compilierung/Interpretierung
Sourcecode: -> Quelltext

Auf den Türmen von Babylon

Im Laufe der Zeit hat sich ein munteres Sprachen-Babylon entwickelt. Neben den Allround-Sprachen wie zum Beispiel BASIC, Modula-2/Oberon oder C, existieren interessante Exoten, wie das reich beklammerte "Scheme" oder das mit der geliebten umgekehrten polnischen Notation gesegnete "Forth".
Manche Sprachen dienen eigentlich nicht als einsatzfähige Sprachen, sondern für Lehrzwecke – zum Beispiel das oben erwähnte "Scheme" als Anschauungsmaterial für den internen Aufbau eines Interpreters. Auch "Pascal" war ursprünglich als Lehrsprache gedacht.
Die Suche nach Sprachen für Spezialanwendungen hat ebenfalls interessante Blüten getrieben. Das oben erwähnte "Forth", gedacht für Steuerzwecke in der Astronomie, ist eines der harmloseren Beispiele. Sogar Programmiersprachen für die Entwicklung von Adventure-Spielen existieren. Sprachen für "künstliche Intelligenz" wie zum Beispiel "Prolog", sind in letzter Zeit im Sektor Personal Computer nicht mehr im Gespräch.
Dabei gibt es auch innerhalb einer Sprache diverse "Dialekte", die je nach Interpreter oder Compiler unterschiedlich sind. Gerade BASIC ist von diesem "Horizontal-Babylon" betroffen. Da BASIC eine ideale "Einsteigersprache" ist, konnte es sich bis vor einem Jahr keine Firma leisten, einen Rechner ohne einem BASIC-Interpreter auszuliefern, bei früheren Heimcomputern sogar im ROM. Daran erinnert eine lustige Runde von BASIC-Dialekten; vom Antik-Basic des ZX-81, (das immerhin schon einen vollen Sprachschatz enthielt), über das Peek-Poke-Basic V2.0 des VC-20/C-64 ("Peek" ist ein Befehl zum Lesen eines Wertes aus dem Speicher, "Poke" das Gegenteil, nämlich zum Schreiben), dem TI-Basic, dem einige "Peeks" und "Pokes" vom C-64-Basic gutgetan hätten, dem "PRINT-75.3-Absturz-BASIC" der ersten Atari-STs bis hin zum etwas verunglückten AmigaBASIC, angelehnt an das MS-Basic für den Apple Macintosh.
Oft machen auch Mischungen aus Hochsprache (C, Modula-2 oder Oberon) und Maschinensprache Sinn – schnellere, handoptimierte Routinen für zeitkritische oder zeitintensive Routinen, der Rest in einer (langsameren) Hochsprache.
Gute Compiler erzeugen aber inzwischen durch ausgiebige Optimierungen einen so guten Code, das sich die reine Assembler-Programmierung im allgemeinen kaum noch lohnt. Die beinahe unendliche Anzahl von Programmiersprachen und Dialekten hat einen ebenso langen wie erbitterten Streit darum, welche Programmiersprache "die beste" ist.
Diese Frage ist natürlich dumm; die ideale Programmiersprache ist a) vom Anwendungszweck und b) vom persönlichen Geschmack abhängig – warum soll nicht jede/jeder die Programmiersprache vorziehen, mit der sie/er am besten zurechtkommt? ❑

Oliver Wagner (jb)



Programmiersprachen

BASIC und Pascal – leicht und verständlich

Programmiersprachen haben den Nachteil, schwer verständlich zu sein. Es gibt aber Sprachen, die sind "etwas weniger schwer verständlich" – BASIC und Pascal.

Wenn ich so an meine Anfänge im Bereich Computer zurückdenke, so fällt mir auf, daß sich der Grund für den Kauf eines "Micros" in den Jahren grundlegend verlagert hat. War es früher ein Hauch von Abenteuer und die Möglichkeit, sich von der breiten Masse abzuheben, so ist es heute einfach fast eine Notwendigkeit geworden.

Damals, als ich meinen Sinclair ZX81 zu mir nach Hause brachte, gab es nicht diese Massen von Programmen, die man kaufen und gleich nutzen konnte – und erst recht keine Disketten. Der kleine Sinclair war eigentlich für mich ein Versuch, festzustellen, inwiefern er sich von einem programmierbaren Taschenrechner unterscheidet. Und siehe da, er unterschied sich!

BASIC – Learning english

Das erste, was ich feststellte, war das völlig veränderte Aussehen der Befehlssyntax, die mich mehr an meinen Englischunterricht erinnerte als an die nmemonischen Befehle, die auf meinem Taschenrechner für Befehlsabläufe sorgten. Das war also meine allererste Begegnung mit BASIC. Die ersten Gehversuche mit dieser Programmiersprache ließen nicht lange auf sich warten. Das BASIC des ZX81 entsprach in seiner Befehlsvielfalt seinen heutigen Nachfolgern, es war (was aus heutiger Sicht seltsam anmutet) sogar besser als das seines Nachfolgers C64. Allerdings war die Hardware nicht so ausgestattet, wie die Software es verlangt hat, ein "Grafikpixel", der durch den Befehl "PLOT" auf den Bildschirm gebracht wurde, entsprach einem vier mal vier Bildschirmpunkte großen Areal – von "hochauflösender Grafik" war damals noch keine Rede.

Nun hat ja nicht jeder Englischkenntnisse, trotzdem ist BASIC leicht zu erlernen, denn fast alle Befehlswörter prägen sich selbst nach kurzer Zeit ins Gedächtnis ein. Die Befehlssyntax entspricht dabei einfachsten Regeln: "PRINT "Guten Morgen"" zum Beispiel verbindet einen Befehl mit einem Parameter, ohne daß noch spezifiziert werden muß, woher dieser Text stammt. Das ist es nämlich, was der BASIC-Programmierer leicht vergißt (was er ja auch darf): Der Text "Guten Morgen" muß ja irgendwo im Speicher stehen, und der Befehl "PRINT" muß wissen, wo.

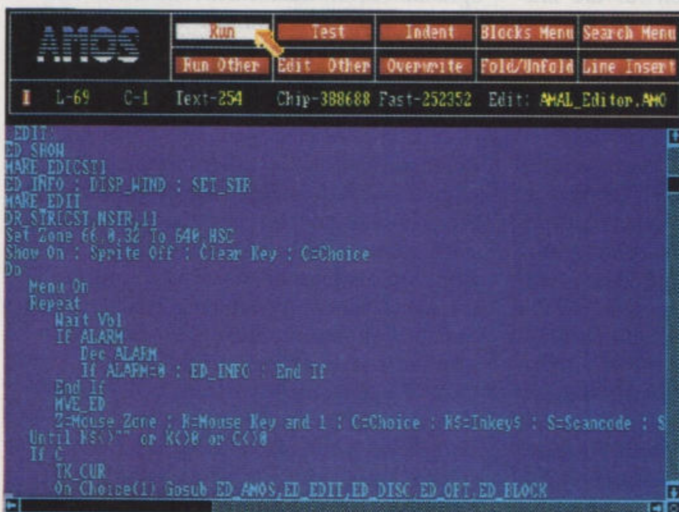


Bild 1. Neue BASIC-Varianten wie »AMOS« erlauben effektive Programme

Das ist eben das Schöne an Sprachen wie BASIC, man braucht sich um die Interna nicht zu kümmern, sondern kann "schnell mal eben was programmieren" – und das ist gleichzeitig das Gefährliche an dieser Sprache.

Das "Schnell-mal-eben-Programmieren" verleitet BASIC-Programmierer dazu, sich an keinerlei Richtlinien zu halten, die man als Programmierer nun mal einzuhalten hat, zum Beispiel, daß Quelltexte (das eigentliche Programm) im Ablauf nachvollziehbar sein sollen. Nehmen wir zum Beispiel ein Anzeigeprogramm für Texte. Fertig ist es, seinen Dienst tut es, wir sind rundum zufrieden. Zwei Monate später sind wir glücklicher Besitzer eines Druckers – und schon möchten wir die angezeigten Texte auch ausgedruckt haben. Kein Problem, wir setzen uns hin, laden unser BASIC-Programm und schreiben eine Druckerroutine, die dann aufgerufen wird, wenn man eine bestimmte Tastenkombination drückt. Nur – wann kann die Kombination abgefragt werden? Und wie? Aus welchen *Variablen* muß ich die Werte retten? Welche Variablen waren das überhaupt, und wo, zum Henker, stehen die noch?

Spaghetti-Codes und wilde Sprünge

Ich mache jede Wette mit Ihnen: Ihre ersten BASIC-Programme werden kaum erweiterbar sein, Sie selbst werden jedesmal von neuem anfangen. Wie gesagt, ich spreche aus eigener Erfahrung. Was da hilft, sind folgende Punkte:

– Machen Sie sich von vornherein Gedanken darüber, was das Programm können und was eventuell später möglich sein soll.

– Schreiben Sie sich eine Variablenliste. Schreiben Sie auf, welche Werte, Parameter oder Aussagen vorkommen. Das geht leichter, als man denkt.

– Fügen Sie an wichtigen Stellen des Programms Kommentare ein. Alle BASIC-Sprachen kennen den Befehl "REM" (vom englischen "Remind"; für den Befehl könnte man es frei mit "Wink mit dem Zaunpfahl" übersetzen). Er dient zum Anlegen von Kommentaren, die vom BASIC selbst nicht beachtet werden.

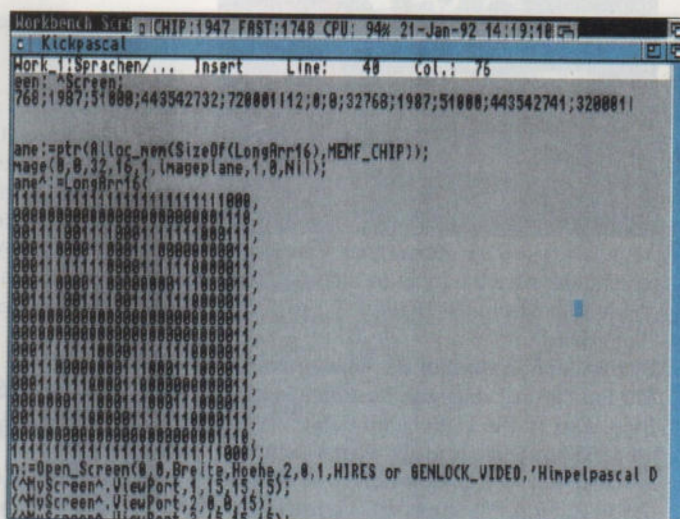
– Planen Sie Erweiterungen gleich mit ein, auch wenn Sie jetzt noch keinen Drucker haben. Jemand anderes, der Ihr Programm nutzen will, wird froh sein, wenn er seine Drucker-Routine an eine vorgemerkte Stelle setzen kann.

– Vermeiden Sie häufiges Hin-und Herspringen zwischen Programmabschnitten. Für jemanden, der noch nicht program-

miert hat, werden die obigen Forderungen böhmische Dörfer sein, doch wer die Gelegenheit hatte, AmigaBASIC auf seiner "Extras-Disk" zu finden, der kann ja versuchen, kleinere Programme nach diesen Gesichtspunkten zu entwickeln.
Doch zurück zum BASIC. Für den Amiga findet man zur Zeit drei gängige BASIC-Varianten. Da wäre zum einen AmigaBASIC, das bis zum Sommer letzten Jahres mit ausgeliefert wurde. Inzwischen findet man es nicht mehr auf den Systemdisketten – ein Grund mehr, sich die beiden anderen BASIC-Varianten anzusehen.
AmigaBASIC ist eine Interpreter-Sprache. Man unterscheidet ja bei Programmiersprachen, genauer gesagt bei Hochsprachen (Assembler ist allerdings damit keine "Tiefsprache"), zwischen Compilern und Interpretern. Interpreter haben gegenüber Compilern eigentlich immer Nachteile, denn:
– Interpreter-Programme (AmigaBASIC-Programme) werden vom Interpreter (AmigaBASIC) Zeile für Zeile abgearbeitet. Zeile für Zeile heißt: Einlesen des Befehls, der Argumente und Parameter, Überprüfen auf Fehler, Abarbeiten des Befehls, weiter zum nächsten. Der Vorteil: Fehler werden meistens schon beim Eingeben erkannt – das war es aber auch schon.
Durch diese Abarbeitung verlangsamen sich Interpreter-Programme so erheblich, daß zeitkritische Abläufe, zum Beispiel Mischung von Sound und Grafik in zeitlichem Verhältnis, nur unter großem Aufwand zu realisieren sind. Wer also Programme plant, die auf solche Abläufe zurückgreifen, der wird Probleme bekommen.
Der zweite große Nachteil ist, daß Interpreter-Programme ihre Arbeitsumgebung erfordern, wenn sie auf anderen Geräten laufen sollen. Wenn Sie also ein AmigaBASIC-Programm, das Sie geschrieben haben, auf Nachbars Amiga zum Laufen bringen wollen, so vergewissern Sie sich, daß der tatsächlich auch den Interpreter besitzt.

BASIC der 90er

AmigaBASIC-Programme können auch compiliert werden, also zu selbstlaufenden, startfähigen Programmen gewandelt werden. Mit dem dazugehörige Compiler habe ich jedoch keine guten Erfahrungen machen können. Zu viele Abstürze und zu viele Fehler, die überhaupt nicht existierten (nicht ordnungsgemäß deklarierte While-Wend-Schleifen wurden erkannt, ohne daß sie existierten) verleideten mir zusammen mit den Unbequemlichkeiten des AmigaBASIC die Lust am Weitermachen. Die Folge dieser Entscheidung war, daß ich dabei wesentliche Möglichkeiten des AmigaBASIC außer acht ließ.
Inzwischen sorgen zwei andere BASIC-Dialekte auf dem Amiga für Furore. Der eine nennt sich »AMOS-The creator«, der andere »Blitz-BASIC«. In beiden Namen steckt ein Teil dessen, was BASIC heutzutage sein kann: einfach in der Programmierung und trotzdem schnell in der Ausführung. Beide Sprachen sind eigentlich Interpreter-Sprachen. Doch sind diese Interpreter auch das, was man sich darunter vorstellt?
Mein Favorit ist »AMOS«. Das "the creator" allerdings zeigt, wo »AMOS« hinzielt: Spieleprogrammierung ist mit diesem Interpreter kein Geheimnis mehr. Der Befehlsschatz von »AMOS« erzeugt Erstaunen. Nicht nur daß strukturierte Programmierung möglich ist, »AMOS« erlaubt sogar Grafikprogrammierung fast wie in Assembler. Und als wäre dies nicht genug, gibt es neben dem Compiler (der zudem noch sehr gut funktioniert) als Erweiterung noch ein 3D-Package, das weitere BASIC-Befehle bereithält, die für dreidimensionale Vektorgrafiken und -animationen stehen. Sieht man sich eine der beiliegenden Demos an, glaubt man, seinen Augen nicht mehr trauen zu können – derart schnellen Vektoraufbau von einem BASIC-Programm hätte man sich gerne vom AmigaBASIC gleich zu Anfang gewünscht.
Wer kein Faible für Spiele hat, ist mit »AMOS« trotzdem gut bedient. Dank ausgeklügelter Dateibefehle sind sequentielle Dateien (gekennzeichnet durch aufeinanderfolgende Daten, die auch nur so eingelesen werden können) ebenso einfach zu handhaben wie "Random-Access-Dateien" (hier sind die Daten "parallel" angelegt, und man kann bestimmte Daten gleich einlesen). Menü- oder Sound-Befehle, Windows oder Screens, mit »AMOS« kann jeder sofort anfangen.
Dasselbe gilt für »Blitz-BASIC«, nur das diese Sprache mehr die "Anwender" anstelle der "Spieler" anspricht, was aber an sich genausowenig richtig ist wie bei »AMOS«. In puncto BASIC machen diese beiden Vertreter die beste Figur. Selbst auf dem PC habe ich kein BASIC gefunden, das dem Programmierer so viel zu bieten hat. Zu Anfang schrieb ich ein wenig über meine eigene Programmier-Historie. Nach dem ZX-81 und dem C64 kam ein PC ins Haus (ich gestehe es ein). PC steht für "Personal Computer", "persönlicher Computer", und so wurde auch ich persönlich und versuchte mich mit ihm zu verständigen. Vom BASIC des C64 auf ein BASIC des PC umzusteigen, kam mir nicht in den Sinn, zumal das einzige mir zugängliche BASIC, »GW-BASIC« genannt, keine bessere Figur machte, als das der "kleinen Brotkiste" von Commodore. Und da ich den ganz ordentlich in Assembler programmieren konnte, experimentierte ich auch ein wenig in dieser Richtung – jedenfalls so lange, bis ich durch einen falschen Funktionsaufruf die Festplattendaten ruinierte.


Bild 2. Pascal achtet streng auf die Einhaltung der Variablentypen

Pascal – erst denken, dann eingeben

Bei der Neuinstallation sah ich in einer Zeitschrift Programme in einer bis dato für mich unbekannten Sprache, die auf den ersten Blick wie BASIC aussah, sich auf den zweiten Blick jedoch als komplizierter erwies. Pascal hieß dieser Versuch, vernünftige Ergebnisse aus Prozessoren herauszufiltern.
Erste Versuche mit dieser Sprache brachten folgende Ergebnisse zu Tage: 1.) BASIC war einfacher, 2.) Pascal war intelligenter.
Brauchte man sich unter BASIC keine großen Gedanken zu machen, wo und wie man ein Programm beginnt, so fängt man in Pascal am besten erst gar nicht an, wenn man nicht weiß, was das Programm alles können soll. Pascal verlangt nämlich am Anfang Typendeklarationen, genauer gesagt, man muß festlegen, ob, wieviel und warum man Variablen benötigt, und dazu



noch, wie groß sie sein sollen. Pascal unterscheidet nämlich nicht in Bit oder Byte, sondern zwischen "Byte", "Word" und "Longword".

Byte ist klar, es besteht aus acht einzelnen Bit. Ein "Word" dagegen besteht aus zwei Byte, also 16 Bit. Dabei sind Zahlen von 0 bis 65535 möglich, die als "Word" bitweise geschrieben werden können (2 hoch 16 = 65536 Zahlen). "Longword" ist noch besser; es besteht aus zwei "Word" oder vier "Byte". Rechnen wir das aus, kommen wir auf 2 hoch 32 = 4294967296 möglichen Zahlen. Pascal verbietet es, Variablentypen miteinander zu vermischen, so sind Realzahl-Variablen (zum Beispiel Pi = 3.1415) tauschbar mit Integer-Variablen.

Eine Word-Integer-Zahl kann nicht mit einer Longword-Integer gewechselt werden. Zwar ermöglichen heutige Pascal-Sprachen durch externe Befehle solch einen Austausch. Der Programmierer muß sich aber trotzdem überlegen, welchen Variablentyp er an welcher Stelle einsetzen will.

Seltsamerweise waren die Nachfolger eher da, als der "Urvater". Gemeint ist folgendes: Pascal, erstellt von dem Schweizer Niklaus Wirth, fühlte sich anscheinend auf allen Computern heimisch, nur nicht auf dem Amiga. Dafür fand man »Modula«, »Oberon« und »Cluster«, die allesamt als Weiterentwicklungen gelten müssen.

Pascal auf dem Amiga

Erst seit einiger Zeit gab es Pascal auch auf dem Amiga. Angefangen hat es mit »PCQ-Pascal«, einem Shareware-Compiler, der das erste Mal ein ANSI-Pascal ("genormtes" Pascal) auf dem Amiga zuließ. Zwar ließ das Handling noch zu wünschen übrig, so mußte der compilierte Code noch durch einen Assembler "gejagt" werden. Trotzdem ließen sich ansprechende und sogar systemnahe Programme erstellen.

Einen richtigen Fortschritt gab es aber erst seit dem Auftauchen von Maxons »Kickpascal«. Hatte sich der PC-verwöhnte "Pascaller" bisher vergeblich auf eine Adaption von Borlands »Turbo Pascal« gehofft, so wurde er jetzt mit diesem Compiler angenehm überrascht.

Eine ansprechende Oberfläche, ein ganz guter Editor, funktionierende Systemprogrammierung (auch wenn das Handbuch sich fast gar nicht darüber ausließ), in der neuen Version das Erscheinen der vom PC her geliebten Units – Pascal-Herz, was willst Du mehr?

Schaue ich mir meine heutigen Programmier-Vorlieben auf dem Amiga an, dann muß ich feststellen, daß sich auch da einiges geändert hat. BASIC hat durch »AMOS« wieder einen gewissen Stellenwert bei mir bekommen, Pascal ist seit »Kickpascal« nun auch auf dem Amiga meine Lieblingssprache, die sie auch auf dem PC war.

Und doch werde ich mich auch mit C beschäftigen (nicht immer, aber immer öfter), und manche Programme werde ich in Assembler schreiben – auch wenn ich dabei dem Guru näher bin. Es kommt eigentlich nur darauf an, was ich schreibe – und da sind Pascal und BASIC bei mir immer noch Spitzenreiter. Aber eben mit ganz kleinem Abstand. ❑

(jb)

Programmiersprachen

Modula und Oberon – Modultechnik

Neben C ist Modula-2 beziehungsweise die Weiterentwicklung Oberon wohl derzeit die einzige Hochsprache, die sich zur Software-Entwicklung auf dem Amiga eignet – behaupte ich hier einmal.

Wie C sind auch Modula-2 und Oberon Compiler-Sprachen, das heißt, der Quellcode muß vor der Programmausführung einmalig in den für den Prozessor verständlichen Maschinencode übersetzt werden. Entwickelt wurden beide Sprachen vom Schweizer Professor Niklaus Wirth an der ETH in Zürich, wobei Modula eine Erweiterung des ursprünglich nur als reine Lehrsprache konzipierten Pascal darstellt. Ziel war eine Sprache, die sich insbesondere zur Programmierung paralleler Prozesse eignet. Oberon wiederum ist aus Modula-2 hervorgegangen und dabei um einige objektorientierte Elemente erweitert worden. So kann in Oberon eine Struktur ("Record") definiert werden, die sich als Erweiterung einer bereits bestehenden Struktur versteht und zu dieser zuweisungskompatibel ist. (Um es etwas einfacher zu definieren: Bestimmte Variablengruppen lassen sich zusammenfassen –"Record" – und sind jetzt in der Lage, sich selbst zu verändern und diese Veränderung an das Hauptprogramm weiterzugeben.)

Um den Compiler-Bau zu vereinfachen, wurden gleichzeitig aber auch einige komplexere Konstrukte entfernt. Beide Sprachen beziehungsweise die darin verfaßten Programme sind modular aufgebaut (lassen sich also Modul für Modul zusammenstellen, wobei die Module unter den einzelnen Programmen austauschbar sind), was auch die Bezeichnung "Modula" verständlich macht. In der Praxis bedeutet dies, daß ein jedes Programm aus beliebig vielen Einzelteilen, den Modulen, besteht. Der Anfang des Programms befindet sich im Hauptmodul, von hier aus werden die verschiedenen Untermodule importiert, was eigentlich nichts anderes bedeutet, als daß der Programmierer dem Compiler mitteilt, aus welchen Modulen er (welche) Prozeduren verwenden möchte. Die Prozeduren sind die eigentlichen Arbeitstiere. Sie werden mit exakt zu definierenden Werten "gefüttert" und liefern ebenfalls definierte Ergebnisse zurück beziehungsweise veranlassen nach außen sichtbare Aktionen, zum Beispiel die Ausgabe einer Information für den Anwender. Werden Module oder die darin enthaltenen Prozeduren allgemein genug konzipiert, können sie ohne Änderung in verschiedenen Programmen Verwendung finden. Ein Beispiel dafür sind die bei den Compilern mitgelieferten Standardbibliotheken, die unter anderem wichtige Funktionen zur Ein- und Ausgabe von Zeichen oder zur String-Bearbeitung beinhalten. Eine umfangreiche Modulbi-

bliothek ist das beste, was einem Programmierer passieren kann.
Der wichtigste Unterschied der Wirthschen Sprachen zu C ist wohl die sehr strenge Typenkontrolle bei der Variablenzuweisung. Variablen sind nur dann zuweisungskompatibel (sprich: können nur dann untereinander Daten austauschen), wenn sie wirklich vom gleichen Typ sind. Variablentypen sind noch lange nicht identisch, wenn sie gleich lang sind. Dies zwingt den Programmierer dazu, sauberen Code zu schreiben, und führt somit am Ende zu einem (hoffentlich) fehlerfreieren Programm.
In deutschsprachigen Raum wurde der Bereich Modula und Co. bis zur Kölner Messe 1990 eigentlich nur von einem Compiler beherrscht: dem »M2Amiga« der Schweizer A+L AG in den Versionen 3.2 und später 3.3. Die beiden zu der Zeit bereits existierenden TDI- und Benchmark-Modula-2-Compiler sind hierzulande nie zu dem Bekanntheitsgrad gekommen, den zumindest der Benchmark-Compiler in den USA erlangen konnte. Wohl nicht zuletzt, weil beide Systeme bis heute ebenso wie das ein wenig später hinzugekommene »M2-Sprint« nicht weiterentwickelt wurden, kann man behaupten, daß sie inzwischen in der Versenkung verschwunden sind. Aber trotz der relativ weiten Verbreitung waren die alten M2Amiga-Versionen nicht unbedingt ein Meisterstück des Compiler-Baus. Der erzeugte Code funktionierte beispielsweise weder mit der »WShell« (einer "Ersatz-Shell" unter AmigaDOS) noch unter OS 2.04 hundertprozentig. Die zweite große Schwachstelle war der Linker, der zum einen das Standard-Amiga-Objektformat nicht verarbeiten konnte und zum anderen nicht optimierend gearbeitet hat, das heißt, wenn aus einem Modul auch nur eine einzige Funktion benötigt wurde, so ist diese direkt in das komplette Modul eingebunden worden. Ein Umstand, der die "Executables" (lauffähigen und startfähigen Programme) unnötig aufgebläht hat.

Konkurrenz oder Nachfolger

Gute Chancen für eventuelle Konkurrenz waren also durchaus gegeben. Und die kam dann auch in Form der beiden Produkte »Cluster« von StoneWare und »Amiga Oberon«, ebenso von der A+L AG. Während das Oberon-System in der Funktionsweise stark an den »M2Amiga« angelehnt ist, geht »Cluster« sowohl in Sachen Bedienung als auch bei der Sprachdefinition eigene Wege. Es handelt sich um eine eigene Erweiterung des Modula-2-Sprachumfangs. Kommt es dem Autor eines Cluster-Programms also unter anderem auf die Portierbarkeit seines Programms an, muß er darauf achten, daß er nur Sprachelemente verwendet, die auch in Modula-2 vorkommen. Da ein gutes Amiga-Programm in der Regel aber ohnehin nicht sonderlich gut portierbar ist, fällt dieses Argument dann doch nicht so schwer ins Gewicht. Bei »Cluster« handelt es sich zudem um ein integriertes Entwicklungssystem, das heißt, Editor, Compiler und Linker befinden sich unter einem Dach. Leider verhält sich die zugehörige Benutzeroberfläche (die in den Editor integriert ist) fast schon konträr zu dem, was man auf dem Amiga gewohnt ist und was im "Amiga User Interface Style Guide" (Standard-Nachschlagewerk für Programmierer) propagiert wird. Auf gut deutsch: So praktisch eine integrierte Oberfläche sein kann, so schlecht bedienbar ist sie möglicherweise.
Trotzdem waren beide Systeme, also sowohl »Cluster« als auch Oberon, dem »M2Amiga« zum Zeitpunkt ihres Erscheinen eindeutig überlegen. Der erzeugte Code war einfach kürzer und schneller. Pünktlich zur 91er Expo in Berlin erschien dann die bis heute aktuelle Version 4.0 des »M2Amiga«, die alle schwerwiegenden Nachteile des Modula-2-Systems aufhob. Ein halbes Jahr später in Köln gesellte sich sogar noch der Pre-Compiler »M2OP« hinzu, der es auch Modula-Programmierern erlaubt, objektorientiert zu arbeiten.
Gleichzeitig mit »M2Amiga 4.0« stellte A+L einen Source-Level-Runtime-Debugger (Fehlersuche wird damit direkt im Quelltext möglich gemacht) zu Oberon vor; eine ungeheuer praktische Erweiterung, die bis heute keines der beiden anderen Systeme vorweisen kann. Lediglich Anwender des inzwischen uralten »Benchmark-Modula-2« dürfen sich ebenfalls zu den Glücklichen zählen. Der zum »M2Amiga« erhältliche Debugger hingegen tritt erst dann in Aktion, wenn das Kind schon in den Brunnen gefallen ist, sprich ein Laufzeitfehler aufgetreten ist oder ein vom Programmierer zu setzender Breakpoint (erzeugt einen vom Programmierer vorgesehenen Abbruch) im Weg stand. »Cluster« verfügt über gar keinen derartigen Zusatz. Leider kann dieses System derzeit auch nur Code für 68000er Prozessoren erzeugen, ein Nachteil gegenüber dem »M2Amiga« und »Amiga Oberon«.
Zum Schluß noch ein kleiner Tip: Eine wahre Fundgrube für den Modula- beziehungsweise Oberon-Programmierer stellt die FD-Serie »Amok« dar. Sie beinhaltet nur Programme, zu denen Modula- oder Oberon-Sourcecode vorliegt. Es gibt also viel zu lernen, sprich abzugucken. Zudem finden sich auf den bis heute über 60 Dis-

```
DF0:QUELLCODE/SerialSupport.mod
PROCEDURE OpenSerial * (Flags: SHORTSET): INTEGER;
  VAR
    Result: INTEGER;
  BEGIN
    ng.Assert (NOT Open, OpenedTwice);
    Result:= notOpen;
    ReplyPort:= es.CreatePort (NIL, 0);
    IF ReplyPort#NIL THEN
      SerRequest:= es.CreateExtIO (ReplyPort, s.SIZE (ser.IOSerial));
      IF SerRequest#NIL THEN
        TimeRequest:= es.CreateExtIO (ReplyPort, s.SIZE (t.IOTimer));
        IF TimeRequest#NIL THEN
          SerRequest^.serFlags:= Flags;
          TimeRequest^.node.command:= t.addRequest;
          TimeRequest^.time.secs:= 10;
          TimeRequest^.time.micro:= 0;
          TimeOutMode:= TRUE;
          IF e.OpenDevice (t.timerName, t.vBlank, TimeRequest, LONGSET{})=0 THEN
END;
          IF TimeRequest^.node.error = 0 THEN
            IF e.OpenDevice (ser.serialName, 0, SerRequest, LONGSET{})=0 THEN
END;
            Result:= SerRequest^.ioSer.error;
            IF Result = serOk THEN
              Open:= TRUE;
              RETURN serOk
            END;
            e.CloseDevice (TimeRequest);
          END;
          es.DeleteExtIO (TimeRequest);
```

Bild 1. Der Modula-2-Compiler von A+L

```
DF0:QUELLCODE/mops.mod
    Lookup(ExtFile, FileName, 20000, FALSE);
    IF (ExtFile.res = done) THEN
      FOR i := 0 TO 63 BY 1 DO
        ReadBytes(ExtFile, ADR(SingleData[i][0]), 131, actual);
      END (* FOR *);
      FOR i := 0 TO 63 BY 1 DO
        ReadBytes(ExtFile, ADR(MultiData[i][0]), 77, actual);
      END (* FOR *);
      ReadBytes(ExtFile, ADR(DrumData[0]), 682, actual);
      FOR i := 0 TO 31 BY 1 DO
        ReadBytes(ExtFile, ADR(EffectData[i][0]), 35, actual);
      END (* FOR *);
    ELSE
      PrintMessage("M.O.P.S. konnte Datei nicht öffnen!", 1,
                   TRUE, TRUE);
    END (* IF *);
    Close(ExtFile);
  END (* IF *);
END ConfLoad;

PROCEDURE SendConfig; (* Den ganzen Müll zum Synthie *)
BEGIN (* SendConfig *)
  ClearSerial;
  PrintMainText("Sende Daten!  ", 23, 1, NORM);
  ClearSerial;
  BuffArray[0] := 240;
  BuffArray[1] := 64;
  BuffArray[2] := 0;
  BuffArray[3] := 34;
  BuffArray[4] := 0;
```

Bild 2. »Cluster« ist ebenso eine Weiterentwicklung von Modula ...



ketten viele Module, die auch in eigenen Programmen verwendet werden können. Zum »M2Amiga« gibt es bei A+L zusätzlich die "Treasures-Serie", die ebenfalls Module zum Weiterverwenden bereitstellt.

WarumOberon?

Ich persönlich verwende Oberon aus einem ganz subjektiven Grund: Ich halte es einfach für lesbarer als C. Da mir Oberon neben C alle Möglichkeiten gibt, die ich zum komfortablen Programmieren des Amiga benötige, habe ich mich einfach für die Sprache entschieden, die mir persönlich mehr zusagt.
Für meine Begriffe ist die Modula- beziehungsweise Oberon-Syntax viel näher an der natürlichen Sprache als C, weshalb es mir wesentlich leichter fällt, mich in die Funktionsweise eines fremden (oder auch eines eigenen älteren) Oberon-Programms hineinzudenken. Zudem zwingt mich Oberon zu einem strukturierten und sauberen Programmierstil. Erst einmal läßt dies den einen oder anderen Flüchtigkeitsfehler gar nicht erst entstehen, zum anderen kann ich mir dadurch relativ sicher sein, daß ein Sourcecode, der sich compilieren läßt, nur noch logische Fehler enthält. Für Oberon und gegen Modula spricht für mich die Existenz des Source-Level-Debuggers und der etwas günstigere Preis des Systems.
Eine Nachteil von Oberon muß man aber trotz allem anerkennen: C hat sich auf dem Amiga durchgesetzt und ist Sprache Nummer eins. Sourcecode-Generatoren (Programme, die die Ergebnisse direkt in die jeweils benötigte Sprache umwandeln, zum Beispiel Window-Daten) wie »PowerWindows« erzeugen C-Sourcen. Die ROM-Kernel-Manuals (die offizielle Dokumentation zum Betriebssystem), tausend Beispiele in Zeitschriften und auf den Fish-Disks, die wenigen guten Bücher als schriftliche Unterlagen – alles ist auf C zugeschnitten. Irgendwie scheint man als Amiga-Programmierer nicht um C herumzukommen. ❑

Christoph Teuber (jb)

```
DF0:Module/AVL.mod                                              7 of 14
PROCEDURE Remove* (VAR root: Root;
                   VAR node: NodePtr): BOOLEAN;

(* Entfernt node aus dem Baum. Ergebnis ist FALSE, wenn node nicht in
   Baum enthalten war. Der Speicher von node muß danach noch freigegeben werden.
*)

VAR
  h: BOOLEAN;
  deleted: BOOLEAN;

  PROCEDURE BalL(VAR n: NodePtr);
  VAR n1, n2 : NodePtr;
      b1, b2 : INTEGER;
  BEGIN
    CASE n.bal OF
    left:  n.bal := 0 |
    ok:    n.bal := right; h := FALSE |
    right: n1 := n.r; b1 := n1.bal;       (* rebalance *)
           IF b1 # left THEN              (* single RR rotation *)
             n.r := n1.l; n1.l := n;
             IF b1 = ok THEN n.bal := right; n1.bal := left; h := FALSE
                   ELSE n.bal := ok;    n1.bal := ok;                END;
             n := n1;
           ELSE                           (* double RL rotation *)
             n2 := n1.l; b2 := n2.bal;
             n1.l := n2.r; n2.r := n1; n.r := n2.l; n2.l := n;
             IF b2 = right THEN n .bal := left  ELSE n .bal := ok END;
             IF b2 = left  THEN n1.bal := right ELSE n1.bal := ok END;
             n := n2; n2.bal := ok;
```

Bild 3. ...wie Oberon, die objektorientierte Sprache

Programmiersprachen

Das Profi-Trio

Neben ABasic und AmigaBASIC war C die erste Sprache, die für den Amiga verfügbar war. Teile seines Betriebssystem wurden damit erstellt, und auch heute wird es von vielen Programmierern wegen seiner Flexibilität sehr geschätzt.

Entwickelt hat sich C aus der Sprache "BCPL" ("basic combined programming language"). Ken Thompson von Bell Laboratories entwickelte hieraus die Sprache "B". Beide Sprachen haben eines gemeinsam: Sie kennen keine Variablentypen (int = integer, char = Zeichen, long = Longword = Vier-Byte-Wert etc.).
1972 erweiterte Dennis Ritchie "B", danach wurde diese Sprache in "C" umbenannt. Zu dieser Zeit enthielt C nur Teile, die zur maschinennahen Programmierung der verwendeten PDP-11-Rechner von DEC (Hochleistungs-Computer) nötig waren. Später wurde C dann um Sprachelemente erweitert, die nicht mehr so systemabhängig waren, womit der Grundstein für ein flexibles Programmiersystem gelegt wurde, welches auch auf Nicht-PDP-Rechnern laufen würde.
Durch seine Qualitäten wurde C zur Standard-Sprache für das Betriebssystem Unix (Mehrplatz-Betriebssystem, das inzwischen auch auf einem speziellen A3000 zu Verfügung steht – siehe auch AMIGA DOS 9/91) und hat sich bis jetzt auch als solche gehalten. Dabei wird im Unix-Bereich hauptsächlich der "GNU C Compiler" verwendet, der – obwohl PD/Freeware – besseren Code erzeugt als die meisten Compiler, die bei den Systemen mitgeliefert werden.
Da C für den Amiga die erste "richtige" Sprache war, war es bis zum Erscheinen von Modula-2 auch die einzige, die sinnvoll zur Programmierung des Systems eingesetzt werden konnte. Mittlerweile sind zwar auch andere gute Hochsprachen verfügbar (Modula-2, Oberon), aber die meisten "Freaks" aus der Anfangszeit des Amiga werden nur selten umsteigen.
C ist so etwas wie eine eierlegende Wollmilchsau. Oder wie es als Vergleich mit einem Auto beschrieben wird: ein Geländewagen, mit dem man durch alles durchkommt, aber sich unter Umständen ganz schön dreckig macht. Die Sprache ist extrem flexibel. Hier kann man mit Pointern rechnen, einen Byte-Pointer in einen Struktur-Zeiger umändern und vieles, ja fast alles, mehr (man versuche mal, unter Modula mit einem Pointer zu rechnen ...). Es gibt sogar Wettbewerbe, bei denen es darum geht, das kryptischste C-Programm zu erstellen (man ist nur dann ein "wahrer" Sieger, wenn nach dem Wettbewerb eine neue Regel festgelegt wird, die das eigene Programm verbieten würde ...). Doch das ist auch das Problem: Da fast alles erlaubt und möglich ist, kann der Compiler nicht feststellen, wenn man etwas aus versehen falsch gemacht hat. Das Ergebnis davon dürfte in den meisten Fällen ein satter Absturz sein.
Mit den aktuellen Versionen der gängigen Compiler ist diese Gefahr jedoch schon etwas reduziert, da man hier verschiedene Typen-Prüfungen und anderes benutzen

kann, so daß einfache Fehler zumindest angewarnt werden.
In den Anfangszeiten des Amiga, als man schon auf ein 512-KByte-System mit einem Laufwerk stolz war, hatte man es als C-Programmierer noch extrem schwer. Die Installation von »Lattice 3.x« auf einer Diskette, bei der dann noch etwa 50 bis 80 KByte Platz frei waren, galt als ein kleines Wunder. Abhilfe schaffte dann einige Zeit später Manx »Aztec C«, welches auch auf einem solchen System problemlos zu verwenden war.
Heute neigt man (angesichts meines 10-MByte-RAM-3000ers mit 200 MByte Plattenkapazität oder den mittlerweile durch Festplatten unterstützten 2000ern) dazu, über diese Zeiten nur noch mitleidig zu lächeln. Die "heutigen Anfänger" haben es ja schon "viel leichter", immerhin hat jeder fabrikneue Amiga 500 plus schon ein MByte Speicher und ist mit weniger als 400 DM mit einem weiteren MByte und einem externen Laufwerk auszustatten. Und damit erhält man ein System, mit dem recht bequem programmiert werden kann.

Wenn C schwierig ist ...

Zur Zeit gibt es für den Amiga drei gängige und brauchbare C-Compiler. »Lattice/SAS C« (aktuelle Version 5.10b) wird zur Zeit noch mit zwei Handbüchern und sechs Disketten geliefert.
Auf diesen befinden sich neben den alten 1.3-Includes auch die Includes zum neuen 2.0-Betriebssystem. Neben dem eigentlichen Compiler und diversen Tools gehört auch ein guter Debugger (sowohl Source-Level als auch Assembler) zum SAS-C-Paket. Außerdem ist schon jetzt die ANSI-Library verfügbar, mit der die Implementierung von Standard-C-Code auf dem Amiga leichter fällt, da viele auf Unix oder anderen Systemen verfügbaren "Standard-Routinen" dort enthalten sind. Ab der nächsten Version 6.0 gehört dies zum normalen Lieferumfang.
Der zweite "Urahn", »Aztec C«, ist zur Zeit in Version 5.0d verfügbar. War in früheren Zeiten die Verteilung der verschiedenen Compiler-Pakete bei den Programmierern etwa ausgeglichen, so hat Manx doch bei der Erstellung der 5.x-Version von »Aztec C« seinen Benutzern keinen Dienst erwiesen. Viele kleine Fehler sorgen dafür, daß größere Programme leicht Probleme machen (Beispiel: Copper-Listen-Fehler in »Railroad Tycoon«). Da auch leider in der letzten Zeit keine Updates oder Bug-Fixes herausgegeben wurden, wird es wohl für Manx sehr schwer werden, seinen Marktanteil zu halten.

Nachdem diese beiden kommerziellen Konkurrenten über lange Zeit unangefochten waren (»PDC«, »Small C« und »Sozobon-C« waren relativ unbrauchbar), haben sie mit Matt Dillons »DICE« jetzt einen starken Gegner gefunden. Dank des unangefochten niedrigen Preises (50 US$ für den Compiler und ein Update; demnächst wird es wohl auch die Möglichkeit geben, »DICE« in Deutschland registrieren zu lassen) sowie einer frei kopierbaren PD-Version ist der Compiler nicht nur für fortgeschrittene Programmierer interessant, sondern auch für den Einsteiger, der nur mal reinschnuppern möchte. Da es zu »DICE« aber leider noch keinen Debugger gibt, muß man sich anderweitig bedienen (wobei es auch hierfür passende PD-Alternativen gibt). Lediglich ein Source-Level-Debugger wird dann noch dringend benötigt.
Will man es noch schneller als im Geländewagen C haben, so muß man im Geländewagen "Assembler" Platz nehmen, der leider ohne jeden Überrollbügel geliefert wird.
Assembler an sich ist eigentlich gar keine Sprache. Angefangen hat es in den Zeiten, als die Maschinen noch mit Lochkarten oder Hex-Eingaben programmiert wurden. Da es mit der Zeit doch recht aufwendig ist, ständig Werte wie "A0 C0 20 D2 FF" einzugeben, hat man sich überlegt, wie man das vereinfachen könnte. Zu diesem Zweck gab man den Befehlen Namen (Mnemonics) und ermöglichte so eine komfortablere (und vor allem besser zu lesende) Eingabe. Außerdem bekamen die Assembler ein wenig mehr Intelligenz, so daß Sprünge selbst berechnet wurden, Makrofähigkeiten und einiges mehr. Irgendwann sind wir dann bei den Assemblern angelangt, wie sie heute bekannt und üblich sind.
Assembler-Programme sind schnell. Ein gutes Assembler-Programm kann durchaus mehrere Male schneller sein als ein entsprechendes C-Programm. Dies liegt wohl hauptsächlich daran, daß noch niemand einen C-Compiler mit genug Intelligenz ausstatten konnte, um den eingegebenen Sourcecode wirlich zu optimieren. Außerdem bemerkt man als Programmierer eigentlich nie, was für ein Aufwand ein einzelner C-Befehl sein kann. Dies wirkt sich meist auch auf die Länge der erstellten Programme aus.
"Nun gut", mag man sich dann sagen, "schreiben wir halt nur noch Assembler-Programme –Ergebnis: schneller und kürzer." Doch leider haben sie einen Nachteil: kann man kleinere Programme (vielleicht bis so 20 KByte) noch relativ gut programmieren (nein, nicht als Einsteiger), so wird es bei umfangreicheren Programmpaketen langsam schwierig. Irgendwann verliert man mit Sicherheit den Überblick über die verwendeten Register, die Namen der Routinen und/oder Dutzende anderer Dinge. Und erst recht dann, wenn man einen Fehler finden will, wird es auch für erfahrene Programmierer schwierig.

... was ist dann erst Assembler?

Im Gegensatz zu den C-Compilern gibt es im Bereich Assembler eine wahre Flut von Programmierumgebungen – eben den Assemblern. Abgesehen von den Assemblern der beiden kommerziellen C-Pakete möchte ich hier nur drei Assembler vorstellen: »ASM 68k« (PD), »ASM-One« und »Makro 68«. (Diese Auswahl bedeutet keineswegs, daß dies die besten sein müssen, lediglich sind dies drei Assembler, die ich persönlich kenne und benutzt habe.)
Beide Assembler von Manx und SAS sind sowohl von der Leistung als auch der Geschwindigkeit her gute Produkte. Während der Aztec-Assembler für die Assemblierung der aus den C-Programmen erzeugten Daten benutzt wird (und man dabei direkt im C-Source auch Assembler-Code einbauen kann), ist der vom SAS-C-Paket eigentlich komplett unabhängig. Beide eignen sich für kleine und große Programme gleichermaßen.

```
DF0:examples/speechtoy.c                                   26 of 46
#endif
        MyCleanup();
        exit(FALSE);
    }

    /* Open the device */
    if ((NarratorOpenError = OpenDevice("narrator.device", 0,
     &voice_io, 0)) != 0) {
#ifdef DEBUG
        printf("Can't open the narrator device\n");
#endif
        MyCleanup();
        exit(FALSE);
    }
    /* This is where the proportional gadgets are set up, using
     * the templates that were declared staticly.
     */
    for(PropCount = 0; PropCount < NUMPROPS; PropCount++) {
        PTexts[PropCount] = TPropText;
        Props[PropCount] = TPropGadget;
        PInfos[PropCount] = TPropInfo;
        Props[PropCount].GadgetText = (struct IntuiText *)
         &PTexts[PropCount];
        Props[PropCount].GadgetRender = (APTR)
         &PImages[PropCount];
        Props[PropCount].SpecialInfo = (APTR)&PInfos[PropCount];
        switch (PropCount) {
        case 0:
            PTexts[PropCount].IText = "Sample Freq:";
            if (DEFFREQ == MAXFREQ)
```

Bild 1. Ein typisches C-Programm



Will man sich nur erst einmal mit dem Gebiet Assembler beschäftigen, so kann man sich auch den Assembler »Asm68k« besorgen, der als PD erhältlich ist. Zusamen mit dem Linker »ALink« kann man hiermit schon komplette Programme erstellen.
Will man es etwas komfortabler, so kann man sich »ASM-One« zulegen. Hierbei erhält man zu einem günstigen Preis ein komplettes System mit Editor, Assembler/Linker und Debugger. Durch kurze Turn-around-Zeiten und viel Komfort zeichnet sich dieses System aus. Da auch das Amiga-Objekt-File-Format unterstützt wird, kann man die hiermit erstellten Programme auch zusammen mit anderen Programmiersprachen nutzen (eine Funktion, auf die man bei jedem Assembler Wert legen sollte, die aber keineswegs von allen Assemblern auch unterstützt wird). Schade nur, daß dieser Assembler bisher noch keinen speziellen 68020/68030-Code erzeugen kann.
Will man es noch etwas leistungsfähiger, so kann man auf »Makro 68« zurückgreifen. Dieser Assembler von den Programmierern des Disassemblers "ReSource" dürfte im Moment von seinen Möglichkeiten her unerreicht sein. Es werden nicht nur sämtliche Befehle der 680x0-Prozessoren unterstützt, sondern auch die neuere Syntax (zum Beispiel "JSR (-192,a6)" statt "JSR -192(a6)").

ARexx – auch Programmiersprache

Der Amiga hebt sich schon seit seiner Entstehung Mitte der 80er Jahre von den anderen Rechnern seiner Klasse durch ein besseres Betriebssystem ab. Die im Betriebssystem enthaltenen Funktionen entsprechen viel mehr denen von erheblich größeren (und teureren) Rechnern. Einen weiteren Schritt in diese Richtung hat William Hawes auch mit der Entwicklung von ARexx gemacht. Diese Sprache, die bis dahin nur für Großrechner verfügbar war (die Rexx/PC-Version war eher ein Witz als eine Implementation, eine ganze Reihe der Rexx-Befehle waren nicht implementiert), eröffnet ein ganzes Feld neuer Möglichkeiten.
Entwickelt wurde Rexx in den Jahren 1979 bis 1982 von Mike Cowlishaw in über viertausend Stunden in den IBM-Labors in England und USA. Ursprünglich für die IBM 370 entwickelt, wurde diese Sprache recht schnell auch für andere Großrechner verfügbar. Durch den einfachen Sprachkern und die geringe Komplexität war Rexx von Anfang an beliebt und ersetzt inzwischen auf vielen Systemen Sprachen wie BASIC oder »Logo«. Trotz allem hat Rexx viele Möglichkeiten, die sonst nur von komplexen Hochsprachen unterstützt werden. Durch diese Mischung von Einfachheit und Erweiterbarkeit ist Rexx für Anfänger und Fortgeschrittene eine ernstzunehmende Alternative zu anderen Programmiersprachen (eine IBM-Filiale verwendet angeblich ein Rexx-Programm mit über einer Million Rexx-Zeilen).
Fragt man einen Amiga-User, was ARexx ist, wird man in den meisten Fällen eine Antwort bekommen wie: "Ein Programm zur Interprozeß-Kommunikation". Zu diesem Zweck wird ARexx wohl seit seiner Veröffentlichung hauptsächlich eingesetzt.
Nicht daß das direkt falsch ist, aber es stellt doch nur einen kleinen Teil der Sprache dar. Dank verschiedener Libraries (und vor allem dank der Entscheidung Commodores, es in die Betriebssystem-Version 2.0 zu übernehmen) ist ARexx inzwischen eine Sprache geworden, mit der man auf dem Amiga fast alles programmieren kann. Angefangen vom einfachen Shell-Script, über ein Programm zum Steuern von »TeX« per Editor bis hin zu einem Programm mit Intuition-Oberfläche und diversen Systemaufrufen.
Steht man als Einsteiger oder Aufsteiger vor der Wahl, welche Programmiersprache man nehmen soll, so ist man erst einmal ziemlich aufgeschmissen. Ich selbst habe als C64-Besitzer und -Programmierer natürlich erst einmal gedacht, daß ich nach dem Erwerb des ach so tollen A1000 gleich da weitermachen könnte, wo ich auf dem C64 aufgehört habe – bei der Assembler-Programmierung. Selbst vom dicken Handbuch ließ ich mich nicht abschrecken.
Doch gab es da ein mehr oder weniger großes Problem: Mitte 1986 waren nicht gerade viele Programme für den Amiga erhältlich – und schon gar kein »ProfiAss«, wie ich ihn vom 64er gewöhnt war. Also mußte ich vorerst mit AmigaBASIC vorlieb nehmen.
Nachdem ich bei einem Freund gesehen habe, zu was für einem Debakel das 3.02er Lattice C auf einem 512-KByte-ein-Laufwerk-Amiga wurde, hatte ich auch wenig Hoffnung, daß ich in der Richtung der "vernünftigen" Programmiersprachen je-

```
ASM-One V1.02 By Rune Gram-Madsen. Source: GettingStarted.S
        LEA.L   COLORS(PC),A1             ; Pointer to the color list
        MOVEQ   #32,D0                    ; 32 colors to set
        JMP     LOADRGB4(A6)              ; Set the colors

;***** Initialize keyboardroutine ******

KEYB_INIT:
        MOVE.L  $4.W,A6
        LEA     CONSOLE_NAME(PC),A0       ; Pointer to "Console.Device"
        LEA     IO_REQUEST(PC),A1         ; Io Buffer
        MOVEQ   #-1,D0                    ; Flags
        MOVEQ   #0,D1                     ; Unit
        JSR     OPENDEVICE(A6)
        TST.L   D0                        ; An error
        BNE.S   STARTUP_ERROR             ; Error quit !!

        MOVE.L  IO_REQUEST+20,CONSOLE_DEVICE    ; Get console device
        MOVE.L  WINDOW_HANDLE(PC),A0      ; Window Handle
        MOVE.L  $56(A0),KEY_PORT          ; Get this windows keyport
        RTS

;***** Exit keyboard *****

KEYB_EXIT:
        LEA     IO_REQUEST(PC),A1
        MOVE.L  $4.W,A6
        JMP     CLOSEDEVICE(A6)

;****** Open Intution-Library ******
Line:   140 Col:   1 Bytes:   12393
```

Bild 2. Assembler-Befehle sind Umschreibungen des Hex-Codes

```
DF0:Sample.rexx                                   1 of 1
options Failat 100
options results

arg name
if name= ""     then
        do
        print "Keinen System-Namen angegeben!"
        call ende
        end

if open(queuefile,perpath || '.systemqueue','A')=0 then
        if open(queuefile,perpath || '.systemqueue','W')=0 then
        do
        print "File " || perpath || ".systemqueue nicht zu öffnen!"
        call ende
        end

address command "date >ram:datetemp"
i=open(dfile,"ram:datetemp",'R')
thedate=readln(dfile)
parse VAR thedate . dt .
parse var dt 1 day 3 . 4 month 7 . 8 year .
value = year * 365 + day
```

Bild 3. »ARexx« setzt sich immer mehr durch

mals auf einen grünen Zweig kommen würde. Doch schon mit dem 3.2er Aztec C kam dann wieder Hoffnung auf; endlich konnte man vernünftig programmieren. Für die erste Zeit war ich dann damit auch zufrieden, Assembler war aufgrund fehlender Unterlagen auch weiterhin gestrichen.

Doch Assembler mußte sein, denn Fehler in Programmen waren nur mittels eines Assembler-Debuggers zu finden, Source-Level-Debugger gab es zu der Zeit noch lange nicht. Erst nach und nach kamen verschiedene kleine Assembler-Hacks zustande. Nun, irgendwann war dann auch das erste, ganz allein geschriebene größere Assembler-Programm fällig.

Die Qual der Wahl

Bis heute bin ich – trotz anderer Programmiersprachen, die es inzwischen gibt – der Sprache C treu geblieben. Als Hacker der "alten Schule" bin ich es gewohnt, Programme einfach "zu schreiben", ohne mich groß um Struktogramme oder ähnliches zu kümmern. Die starren Regeln und Gerüste, die bei Wirthschen Sprachen vorgegeben sind, sind mir ein Graus. Meine zwei Semester Pascal im Studium liegen auch schon lange hinter mir (glücklicherweise). Und die Apollo Workstations an der FH sind in C auch recht gut programmierbar. Bei der Programmierung meiner Programme greife ich, sofern ich es für sinnvoll erachte, auf Assembler zurück (zum Beispiel "serlib.library").

Wenn es darauf ankommt, daß ein Programm möglichst kurz oder auch möglichst schnell ist, dann ist Assembler bisher noch ungeschlagen, auch wenn es bei der Programmierung etwas mehr Disziplin, Wissen und Können verlangt.

Bei kleineren Dingen, die einfach "mal so" programmiert werden sollen, muß der ARexx-Interpreter beweisen, daß es ihn noch gibt. Sei es ein Install-Script für ein Programmpaket oder ein Steuer-Script, um vom »Cygnus-ED« aus ein anderes Programm zu steuern: ARexx ist in solchen Fällen ein gern gesehener Helfer. Die Programme sind schnell programmiert und stürzen auch bei üblen Fehlern so gut wie nie ab.

Alles in allem glaube ich, daß ich mit der Konbination von SAS C (und seinem Assembler) sowie ARexx wohl auch zukünftig gut gerüstet bin. Und auch jedem Modula-, Oberon- oder Cluster-Programm Paroli bieten kann.



Garry Glendown (jb)

Nicht nur Programmieren ist schwer, auch das Behalten und Kennen der verschiedenen Fachbegriffe bedarf einigen Aufwands. Zur Hilfe finden Sie hier die im Text vorkommenden Fachbegriffe erläutert.

Betriebssystem: Ein "Grundprogramm", das dem Computer erklärt, wie er zu arbeiten hat. Beim Amiga wird zum Beispiel die Hardware überprüft, danach werden die Systemparameter für CLI, Workbench, Windows etc. gesetzt

Variablentypen: Hochsprachen verlangen Unterscheidungen in der Länge der Variablen sowie für den Inhalt. Der Type "ULONG" zum Beispiel deklariert in C ein "Unsigned Longword", also eine Vier-Byte-Variable ohne Vorzeichen.

Hochsprachen: Alle Sprachen, die keine Zahlen- oder mnemonische Eingabe, wie bei Assembler, erwarten. Statt dessen wird eine, meist geläufige, Umschreibung der Befehle benutzt.

Pointer, Byte-Pointer, Struktur-Zeiger: Zeiger (engl.: Pointer) dienen dazu, auf bestimmte Variablen im Speicher zu "zeigen". Wird zum Beispiel die Variable "Wert" in den Speicher geschrieben, so kann man mit dem Pointer "Wert^" den Platz für die Variable festlegen und Ein- und Ausgaben auf den Pointer verweisen. Pointer gibt es für System- und User-Variablen.

Includes: Programmteile, die immer wiederkehrende Aktionen beinhalten, zum Beispiel das Öffnen der "Exec", des Betriebssystems des Amiga. Mit Includes läßt es sich vermeiden, wichtige Programmstrukturen immer neu schreiben zu müssen – man lädt sie vor dem Compilieren nach und compiliert sie einfach mit.

Debugger: "Entwanzer" – hilft bei der Fehlersuche in Assembler-Programmen, indem er zum Beispiel Schritt für Schritt arbeitet und den Inhalt der benötigten Speicherzellen sowie der Prozessorregister ausgibt. "Source-Level-Debugger" sind für Hochsprachen gedacht.

ANSI-Library: Standardisierte Befehlsbibliothek

Implementierung: Sind in einem Assembler ein Speichermonitor und ein Debugger direkt enthalten und aufrufbar, so sind diese implementiert.

Copper-Listen: Der "Copper" ist ein Teil des Grafik-Chips. Er ist zum Beispiel für diese berühmten Farbspiele im Hintergrund zuständig (rollende Farbpalette). Der Copper versteht genau drei Assembler-Befehle. Schreibt man mit diesen Befehlen mehrere Anweisungen, erhält man eine Copper-Liste.

Bug-Fixes: In einem Programm sind Fehler festgestellt worden, die mangels Wissen, Unterlagen, Lust nicht behoben werden können. Meistens gibt man diese Fehler an den Benutzer weiter, mit der Hoffnung, daß der weiterkommt. Bug-Fixes enthalten Informationen über gefundene oder nicht gefundene Fehler und werden weitergegeben.

Mnemonics: Maschinensprache zu programmieren würde in einer Anhäufung von Nullen und Einsen enden. Um die Sache zu vereinfachen, wurden die Befehlsbegriffe genommen, abgekürzt und als "mnemonische Umschreibung" des Befehls eingesetzt.

Makros: Bestimmte Teile eines Programms lassen sich als Makro definieren und wirken wie ein "Pseudo-Befehl", daß heißt, man kann sie so ähnlich wie einen richtigen Befehl aufrufen.

»ALink«: Das Amiga-System-Link-Programm wird gebraucht, um Programmteile (zum Beispiel Assembler-Routine in C-Programm) einzubinden – zu linken.

Turn-around-Zeit: Zeit, die beim Erstellen des Programms anfällt: Schreiben – Compilieren (Assemblieren) – Testen – Debuggen – Korrigieren – zurück zum Anfang. Integrierte Softwarepakete haben alle benötigten Programmumgebungen zusammengefaßt, dies spart einiges an Erstellungszeit.

Amiga-Objekt-File: Fertiges, compiliertes Programm, daß jedoch keine Startfähigkeit besitzt. Das Objekt-File kann in andere Programmteile eingebunden ("gelinkt") werden.

IBM 370: Großrechenanlage für Firmen, zum Beispiel Versicherungen

Logo: Die Ninja-Turtles sind nicht zuerst dagewesen, denn eine "Turtle" gab es schon vorher auf Computern. Sie war der Grafik-Cursor der Sprache "Logo", die hauptsächlich für Kinder geschrieben worden war, um ihnen den Umgang mit dem technischen Monstrum zu erleichtern. »Logo« ist ein grafisch orientierte Sprache, wobei sich die Koordinaten des Textes ebenfalls auf Grafikkoordinaten bezogen.

TeX: Seitenbeschreibungssprache, ähnlich wie Postscript. Man kann komplette Seiten mit Text und Grafik erstellen, diese werden in ein Computersprachen-ähnliches Format gebracht.

Struktogramme: Programmplanungen, die den Ablauf des Programms schematisch festhalten.

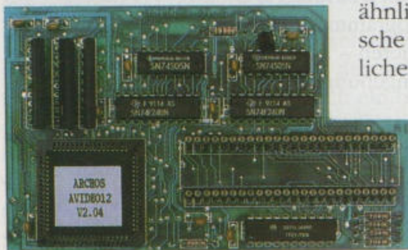
Test

ColorMaster

Klein, aber oho

Luxus bei niedrigen Kosten, das ist eigentlich ein Widerspruch. Doch es geht auch anders, wie die neueste Grafikkarte des Münchner Distributors bsc beweist.

Bei »ColorMaster« handelt es sich um einen weiteren Framebuffer, ein Hardware-Zubehör, das sich auf die Darstellung von IFF-Bildern spezialisiert hat, die vom hauseigenen Grafiksystem des Amiga wegen zu vieler Farben nicht mehr gezeigt werden kann. Mit maximal 4096 Farben ist der Amiga von der Grundausstattung her gar nicht so schlecht bestückt, die Ansprüche sind jedoch schon seit geraumer Zeit deutlich über diese Grenze hinausgewachsen. Echtfarben, ein Schlagwort der elektronischen Bildverarbeitung, müssen es für den Grafik- und Video-Fan schon sein, und für diese benötigt der Computer pro Bildpunkt 24 Bit (16,7 Millionen Farben). Zum Vergleich: Der Amiga stellt im "normalen" Grafikmodus gerade 6 Bit tiefe Pixel dar, was für 64 Farben reicht (2 hoch 6 = 64). Nur mit Tricks sind 4096 (HAM; hold and modify) Farben möglich, dabei ist aber die Auflösung der Grafik auf 368x580 Pixel beschränkt.



Die meisten Framebuffer arbeiten nach dem Prinzip, daß zwar die Daten vom Amiga übernommen werden, deren Darstellung jedoch auf einem zusätzlichen Monitor erfolgt. Ein Umstand, der zum ehemals revolutionären Grafik-Computer Amiga nicht so recht passen mag. In der 24-Bit-Version beherrscht der »ColorMaster« diese Disziplin ebenfalls, geht aber zum Teil auch eigene Wege. Nicht nur, daß dieses Grafik-Subsystem keinen der Erweiterungssteckplätze eines Amiga 2000 belegt, es ist sogar wesentlich kleiner. Allein dieses technische Konzept läßt erahnen, daß zumindest von den Ausmaßen her auch ein Einsatz im Amiga 500 möglich ist. Tatsächlich kann laut Handbuch das "Kärtchen" in jedem Amiga Verwendung finden, also auch im Amiga 3000, obendrein steht die erzeugte Bildinformation für das Mischen mit Genlocks bereit. Für Spiele-Liebhaber: Die Hardware fügt sich so nahtlos in das System ein, daß Spiele weiterhin lauffähig bleiben.

Justieren ohne Schraubenzieher

Der Einbau ist für alle Amiga-Typen gleich – man öffnet das Amiga-Gehäuse, hebelt unter Beachtung der elektronischen Vorsichtsmaßnahmen (vor den Arbeiten eine Heizung oder ähnliches berühren, um elektrische Entladungen auf empfindliche Chips zu vermeiden) den Denise-Baustein aus seinem Sockel. Dieser wird in den Steckplatz der »ColorMaster«-Karte eingesetzt, die anschließend den Platz auf der Hauptplatine einnimmt. Damit Kurzschlüsse durch Kontakt zu Bauteilen der Platine vermieden werden, legt der Distributor jedem Paket zusätzliche Sockel bei. So kann der erforderliche Sicherheitsabstand für jedes System mit wenigen Handgriffen eingestellt werden. Nach dem Zusammenbau des Rechners erfolgt die Justierung der Karte. Dies ist kein Tippfehler, im Gegensatz zu vielen anderen Subsystemen (zum Beispiel: Flickerfixer), kommt der »ColorMaster« für die Grundeinstellung mit Software aus. Nach dem Start der Steuersoftware (ohne diese gibt der »ColorMaster« kein Pixel von sich) wird ein Justierprogramm aufgerufen, das durch einfache Mausklicks eventuell vorhandenes Farbrieseln beseitigt. Der dabei gewählte Wert wird bei jedem Systemstart durch ein Kommando eingestellt.

Kernstück der Huckepack-Platine sind Speicherbausteine (4 MByte Zip-RAMs), die entsprechende Datenmengen aufnehmen können – 1,5 MByte sind auf dem »ColorMaster 12« installiert. Die eigentliche Arbeit leistet nach wie vor der Denise-Chip des Amiga, nur daß ihm von der »ColorMaster«-Elektronik andere Daten untergeschoben werden. Um diese auch auf dem Amiga-Monitor zu Gesicht zu bekommen, muß man lediglich die Farbpalette der Workbench anpassen. Die Farbenpracht wird ähnlich dem Genlock-Prinzip in das erste Farbregister eingespielt, sichtbar wird sie, wenn dieses auf der Workbench auf Schwarz gestellt ist. Zitat des Herstellers: "Dadurch ist es möglich, im

Listing. Das nachfolgende ARexx-Script wird von »ADPro2« verwendet, um intern erzeugte 24-Bit-Daten durch den »ColorMaster« anzuzeigen (im Verzeichnis AREXX: speichern). Es kann sowohl auf eine der neun F-Tasten von »ADPro« gelegt, als auch durch die F10-Taste per File-Requester aufgerufen werden.

```
/* AVDisplay.adpro - ColorMaster-Saver für ADPro2, */
/* Workbench-Register 0 mit */
/* Preferences auf Schwarz stellen — D. Zivadinovic */
/* trace r */
/* Laden der Support Libraries */
check = addlib('rexxsupport.library',0,-30,0)
check = addlib('rexxmathlib.library',0,-30,0)
check = addlib('rexxarplib.library',0,-30,0)
check = addlib('screenshare.library',0,-30,0)

address "ADPro" /* Empfänger folgender ARexx-Kommandos */

options results

adpro_to_front

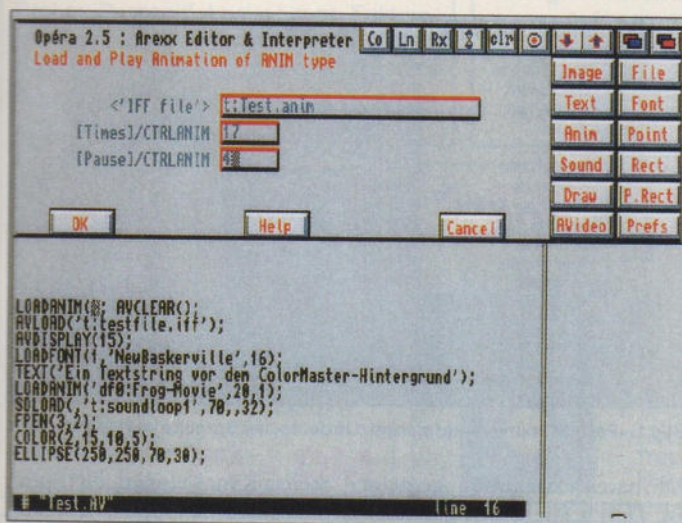
sformat IFF /* Wahl des Speicherformats */
if RC ~= 0 then do
okay1 "Konnte IFF-Modul nicht selektieren"
exit
end


getfile '"Bitte, Dateinamen wählen" "t:" "TESTfile.iff"'
if RC ~= 0 then exit
TheFile = ADPRO_RESULT



save TheFile raw
if RC ~= 0 then do
okay1 "Fehler während der Speicheroperation von" TheFile
exit
end
adpro_to_back /* Screen von ADPro nach hinten schalten */

address 'avideo'; /* neuer Empfänger folgender ARexx-Kommandos */
AVLace 1
AVShow TheFile; /* Datei laden und anzeigen */
result=request(0,0,"Fenster schließen um\ ColorMaster Display zu löschen",,,,)
if result~=0 then /* erst wenn Fenster geschlossen wurde */
AVClear;
/* wird Display gelöscht und ADPro nach vorne geschaltet */
```



Multimedia und ARexx – zwei Welten kooperieren

Hintergrund ein 12-Bit-Bild darzustellen und gleichzeitig im Vordergrund eine Amiga-Animation abzuspielen." Hinzuzufügen bleibt, daß das Hintergrundbild zwar auch "nur" 4096 Farben enthält, jedoch weder an die Auflösungsbeschränkung des HAM-Modus gebunden ist, noch zusätzlichen Rechenaufwand des Amiga erfordert.

Tatsächlich sind Grafiken bis zu 768x575 Größe darstellbar, wobei 24 Bit tiefe Daten von der Hardware automatisch auf 12 Bit umgewandelt werden. Das RAM kann in zwei Bereiche unterteilt werden (Buffer; zu deutsch Puffer). Die Anzeige ist zwischen Buffer und Haupt-Screen umschaltbar – eine Möglichkeit, 24-Bit-Hintergrundbilder rechtzeitig vor einem Animationsereignis zu laden. Zur Einblendung steht eine relativ grobe Hardware-Routine zur Verfügung, hier wären feinere Abstufung und Überblendeffekte wünschenswert.

Die Software

Eine Demo-Version des adaptierten »TV-Paint« ist im Lieferumfang enthalten, daneben ist auch »AVPaint«, ein kleineres Malprogramm, bereits jetzt verwendbar. In Zusammenarbeit mit einem Flickerfixer ergeben sich aber Probleme, da nur jede zweite Zeile angezeigt wird, was zu verstümmelten Darstellungen der Benutzeroberfläche führt. Ein weiterer Unterschied zu anderen Grafiksystemen ist, daß der "Farbmeister" nicht ausschließlich auf angepaßte Software (also auf Drittherstellerl) angewiesen ist. Wie so oft, ist auch hier "ARexx" das Schlüsselwort. Die Steuersoftware (AVideo) akzeptiert eine Vielzahl von Kommandos, die beliebige ARexx-Sender schikken können. Das einfache Beispiel, eine Diashow, läßt sich ohne weiteres über die Amiga-Shell realisieren; noch einfacher, da ohne Programmierung, geht es mit dem beigegebenen Anzeigeprogramm. Hier braucht der Anwender lediglich in einem Dialogfenster den Dateinamen des IFF-Bilds anzugeben, den Rest besorgt die zugehörige Software.

Weit interessanter ist das Zusammenspiel mit »Art Department Professional« (kurz »ADPro« – Bildbearbeitungsprogramm, das intern mit 24 Bit arbeitet). Durch einen Tastendruck können die Originaldaten automatisch vom »ColorMaster« eingespielt werden. Ein ARexx-Script, bestehend aus bescheidenen 39 Zeilen (siehe Listing), erledigt sowohl das notwendige Zwischenspeichern in »ADPro« als auch das Laden und Darstellen durch den Grafik-Winzling. Dieser ist noch so neu auf dem Markt, daß vom Hersteller des »ADPro« noch kein "echtes" Speichermodul hergestellt wurde. Ferner ist im Paket ein weiteres Programm enthalten, das ebenfalls stark auf ARexx baut – »Opera«. Dieser Vertreter der Multimedia-Programme soll über ein eingebautes Hilfslexikon (englisch) verfügen, so daß man nicht ständig mit der Anleitung hantieren müßte – die entsprechenden Dateien waren jedoch auf der Testdiskette ebensowenig enthalten wie ein Tutorial. Ähnlich anderen Konkurrenten dieser Kategorie, bietet auch dieses Programm bekannte Operationen für den Entwurf und das Vorführen von Präsentationen (Abspielen von Soundtracker-Musikdateien, Ein- und Ausblenden von IFF-Bildern, ANIM-Dateien und Texten).

Zahlreiche Effekte und Überblendungen sind über englisch beschriebene Schalter verfügbar, Systemschriften und Größen sind für Textbeiträge frei wählbar. Die gewählte Operation (beispielsweise Bildanzeige über »ColorMaster«) wird automatisch in ein ARexx-Kommando mit zugehörigen Parametern übersetzt und in einem gesonderten Fenster angezeigt (siehe Bild). Die mausorientierte Benutzeroberfläche und der integrierte Editor erleichtern besonders dem Anfänger die ersten Berührungen mit ARexx-Scripts, fortgeschrittene Anwender können entsprechende Makros auch außerhalb des »Opera«-Systems mit einem Editor freier Wahl programmieren.

Bei »Opera« handelt es sich um ein System für Heimanwender, was sowohl der relativ niedrige Preis als auch das Fehlen jeglicher Treibersoftware für so kostspieliges Zubehör wie Bildplattenspieler und CD-ROMs zeigen. Die Mischung aus Tastaturbedienung und Mauseingaben stellt für das gewählte Konzept eine angenehme Hilfe dar, da dennoch viele Operationen numerische Eingaben verlangen (Beispiel: Zeichnen von Ellipsen), braucht man für intuitiven Umgang eine gewisse Einarbeitung, die Benutzerführung bedarf wegen der Zweisprachigkeit (überwiegend englisch, zum Teil auch französisch) einer Überarbeitung. Eine grafische Ablaufanzeige, wie sie beispielsweise der »ShowMaker« hat, würde sowohl die Übersicht als auch die Synchronisation einzelner Ereignisse erleichtern.

Resümee

Sowohl wegen kleiner Ausmaße als auch wegen interessanter Fähigkeiten empfiehlt sich der Bilderanzeiger allen, die mit 24-Bit-Grafiken arbeiten. Wer vor der Programmierung zurückschreckt, wird angenehme Arbeitserleichterungen vorfinden, die auch dem Anfänger den Einstieg erlauben. Die Darstellungssoftware zeigte sich bis auf einen Fehler von positiver Seite (8-Bit-Graustufenbild provoziert Systemabsturz), das Präsentationsprogramm »Opera« weitet die Anwendungsmöglichkeiten auf das Multimedia-Gebiet aus. Günstig ist ebenfalls, daß eine Zusammenarbeit mit Genlocks und Flickerfixern (fast) problemlos ist. Bleibt zu hoffen, daß ein brauchbares Tutorial zum »Opera«-Programm demnächst auch diese Software leichter zugänglich macht. ❑

Dusan Zivadinovic (vb)

ColorMaster 12

AMIGA DOS

Testurteil 3'92

gut	80%
Anleitung:	60%
Installation:	80%
Bedienung:	80%
Tauglichkeit:	90%
Preis/Leistung:	90%

Hersteller: bsc
Testmuster: Hersteller
Preis: ca. 800 DM
Konfiguration: A500, A2000, A3000
Einsatzbereich: semiprofessionell

Perfect Sound 3.2

Noch perfekter?

Bei den professionellen Sample-Programmen gibt es eigentlich zwei, die seltsamerweise immer gleichzeitig in neuer Version auftauchen: »AudioMaster« und »Perfect Sound«. Während ersterer schon Version "IV" erreicht hat, kommt »Perfect Sound« "erst" mit Version 3.2 heraus. Was ist neu am "perfekten Sound"?

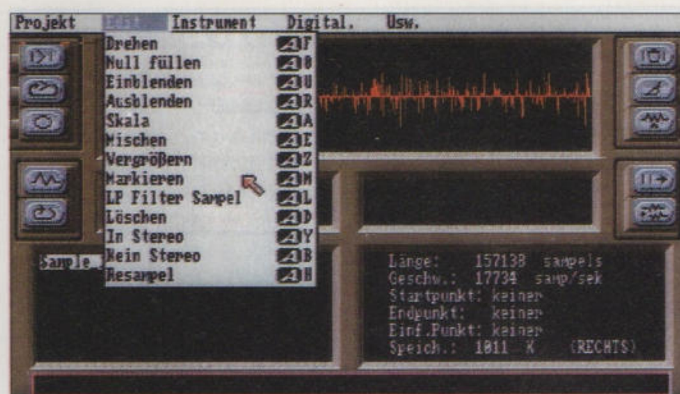


Bild 1. »Perfect Sound« zeigt sich jetzt in deutscher Sprache

Während »AudioMaster« bis jetzt nur als Software ausgeliefert wurde (die Ausnahme war Version "III" mit dem Sampler »SoundMaster«), bilden bei »Perfect Sound« jeweils Audio-Digitizer (beachten Sie bitte hierzu auch den Workshop in diesem Heft) und Sampler-Software eine Einheit.

Die erste Überraschung erwartet den Käufer bei der Installation. Die Datei »Perfect Sound.TXT«, die für den Text der Pull-down-Menüs sowie für alle Informationen innerhalb des Programms zuständig ist, kann bei der Installation wahlweise in deutscher oder englischer Version geladen werden.

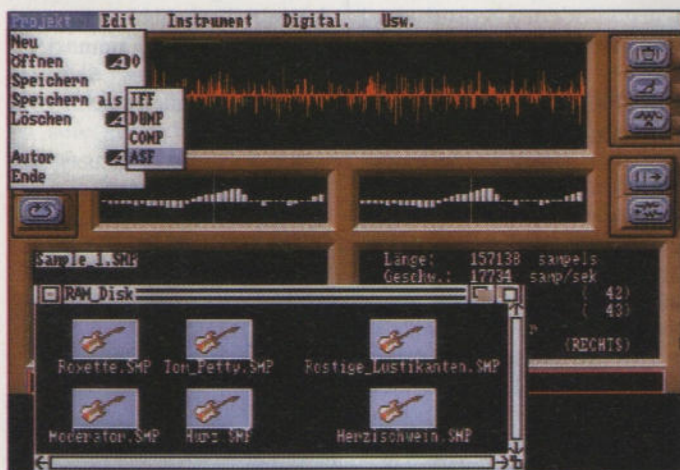


Bild 2. ASF-Files sind Samples, die lauf- und startfähig gemacht wurden

Menüführung in Deutsch

Das deutsche Handbuch ist so ausführlich geschrieben, daß zusammen mit dem Programm keine Mißverständnisse aufkommen können, da bereits die Menüführung aussagekräftig genug ist.

Des weiteren fällt auf, daß die Einstellung der Aufnahmegeschwindigkeit tatsächlich der entspricht, die vom Programm angezeigt wird.

Die uns vorliegende Software, Version 3.22, brachte dabei noch weitere Verbesserungen des Programms mit. So werden jetzt die Byte-Werte des Samples bei der Start- und Endmarkierung zusätzlich angezeigt. Soll ein Teil der Aufnahme herausgeschnitten werden, so sind Start- und Endmarkierung durch das Ziehen des Mauszeigers mit gedrückter linker Maustaste zu setzen.

Ganz neu und interessant ist die Möglichkeit, Samples als "ASF-File" zu speichern. Die Überraschung wird um so größer, wenn man merkt, daß mit "ASF" ein Executable, also ein ausführbares Programm, gemeint ist. Durch Speichern des Samples im ASF-Format läßt sich dieses wie ein normales Programm (zum Beispiel CLI-Kommando) aufrufen und starten. Da das Executable sich auch mit dem Befehl »Run« als neuer Task starten läßt, sind kleinere Sound-Gags in "Startup-Sequences" wohl demnächst Usus — sofern man »Perfect Sound« in der neuen Version besitzt. Zusammen mit einem Project-Icon, zum Beispiel mit »Icon-Ed« erstellt, lassen sich die Samples auch von der Workbench aus starten.

Solide und ausgereift

Neben diesen neuen Funktionen bietet »Perfect Sound« natürlich noch alles das, was ein Sample-Programm nützlich macht. Neben den Schneide-Funktionen lassen sich die Samples drehen, mischen, ein- oder ausblenden, Stereo- oder Mono-Samples können in das jeweils andere Format konvertiert werden. Neben diesen Funktionen bietet »Perfect Sound« auch noch das "Echtzeit-Echo" und die "Echtzeit-Verzögerung" an. "Echtzeit-Echo" legt auf eingelesene Tonkanäle direkt (in Echtzeit) ein Echo an, das sich in weiten Stufen regeln läßt, "Echtzeit-Verzögerung" läßt das eingespeiste Signal um eine vom Benutzer einzustellende Verzögerungszeit später durch den Lautsprecher schallen.

Die Hardware hat sich nicht geändert, neben zwei durch Cinch-Buchsen repräsentierten Stereo-Kanal-Eingängen findet man noch einen Mikroeingang in 3,5-mm-Klinkenbauweise am kleinen, schwarzen "Sampler-Kästchen". Beide Eingänge können über die Software beim Vorhören eingepegelt werden, so daß man gleich mitbekommt, ob Verzerrungen möglich sind oder ob das Signal "irgendwo verschwindet".

Alles in allem bietet das Perfect-Sound-Paket einiges an Leistung für Liebhaber der zusammengeschnittenen Musik. Wer sich trotzdem »AudioMaster« zulegen will, hat Glück: Beide Produkte harmonieren hervorragend miteinander, die Sampler von »Perfect Sound« und »AudioMaster« sind dank Einbindung in die Software ein prima Gespann. ❑

(tb)

Perfect Sound 3.2

AMIGA DOS

Testurteil 3'92

gut	85%
Anleitung:	84%
Installation:	82%
Bedienung:	87%
Tauglichkeit:	85%
Preis/Leistung:	85%

Hersteller: SunRize Industries
Vertrieb: Advanced Technologies
Preis: 149,– DM
Konfiguration: alle Amiga

SUPERPACK 50

50 PD-Programme der Extraklasse!

Return to Earth, Kampf um Eriador, Risk, Broker, Paranoid, Lucky Loser, Faktura, MS-Text, Videodatei, Plattenliste, Superliga, Haushaltsbuch, MCAD, Wizard of Sound, CLI-Pack, Virus-Stop, Werner spiel, Latein, ROM, Star Trek, Core Wars, Label, Amiga-Paint, Giroman, Blizzard, Virus Control, Tetnix, Moria, Battleforce, Peters Quest, Super-Bilder, Billard, Einkommensteuer, DSort III, Fix-Disk, Universaldatei, Quickmenü, Diskey, Mandelbrot, Silver-Bilder, Astronomie, Superprint, Calc, Atlantis, Schach, Labelpaint.

Fast alle Programme mit deutschen Anleitungen

Komplettpreis für alle Programme **79,- DM**

VORSICHT

vor Billigangeboten unserer Programmsammlungen von nicht autorisierten Händlern!
Nur bei uns erhalten Sie die jeweils aktuelle Version mit allen Service-Vorteilen (Support usw.)

PD-SHOP

PUBLIC-DOMAIN/SHAREWARE LOW-COST-SOFTWARE



Das bieten wir:

- geprüfte 3,5"-Qualitätsdisketten
- auf Viren geprüfte Disketten
- versandgerechte Verpackung
- **Einsteigerdiskette mit Tips und Tricks sowie Antivirus-Programm liegt jeder Bestellung bei!**

SPIELESAMMLUNG

NEU! *Jetzt noch mehr Programme!*

Fast alle Programme mit deutschen Anleitungen

Eine Spielesammlung der Extraklasse! Hier ist für jeden etwas dabei, ob Action-, Strategie- oder Gesellschaftsspiel – ein Muß für jeden Spielefan!
Imperium Romanum ist ein Strategiespiel für 2 Spieler, **Pythagoras** eine tolle Handelssimulation für 1-4 Spieler, **Tetris** – bekanntes Spielprinzip mit 2-Spieler-Modus, **Faxen** ein lustiges Puzzlespiel, **Ball + Pipes** eine besondere Variante von "Vier gewinnt", **Hiruris** ein Geschicklichkeitsspiel besonderer Art, **Blox** ordnen Sie herabfallende Steine, **Spacebattle** ein Ballerspiel, **Drive Wars** ein weiteres Ballerspiel, **Disc** Glücksspielsimulation, **Clowyns** ein deutsches Textadventure, **Drip** ein besonders gutes Actionspiel, **Mykene** spannendes Strategiespiel, **Roll On** tolles Labyrinthspiel, **Obsess** eine weitere besonders gute Tetris-Variante, **Paranoids** ein lustiges Gesellschaftsspiel, **SYS** Labyrinthspiel mit lustiger Spielidee, **Miniblast** ein Helicopter-Spiel, **Car** ein Autorennspiel, **Hubert**, lustiges Hüpfspiel, **Glücksrad**, bekanntes Quizspiel, **Hearts + Spades** tolles Kartenspiel
Diese Spielesammlung mit allen aufgeführten Spielen kostet nur **39,- DM**

SUPERPACK II PLUS

Artikel-Nr. PDA002

Ausgesuchte Spitzenprogramme, die für jeden Amiga-Anwender interessant sein dürften: Banner II, Power Packer, Sonix-Player mit Musik, Xytronic, ASDG-Ram-Disk, Fußballmanager, Steinschlag, MR-Backup, Bibel-Quiz, Boulder 1,3, RollOn, Tumbler Street, Labelprint, Thundercopy, Trucking, SchreibM, Hyperadress, Ultrapaint, Quizmaster, Pinball, Database, Workbench-Programme.

NEU im Paket: Spielesammlung mit 10 Programmen für jeden Spielefan: Mensch freu dich doch, Gladiator, Pac-Mac, Brainstorm, Dungeon-Castle, Sculptor, Waltermat, Jumper, Bodorynth, Duell; und 3 unentbehrliche Anwenderprogramme im Wert von 70 DM; Briefkopfdruck mit Text-Editor und Serienbrieffunktion, VideoPro; umfangreiche Videoverwaltung, DSortPro; umfangreiche Programmverwaltung!

Alle Programme mit deutschen Anleitungen!

Komplettpreis für alle Programme nur **79,- DM**

MUSIKPAKET

Intui Tracker – grafisch sehr gut aufgemachter Soundtrackerplayer mit Kontrollinstrumenten, **Sequencer** schnell und einfach mit Samples Musikstücke komponieren, **MED** ein toller Musikeditor, **Beatstompec** simuliert einen Drumcomputer. Außerdem enthält das Paket eine **Riesenmenge ausgewählter Soundtracker-Musikstücke** zum anhören und bearbeiten. Das ideale Paket für jeden Musikfan zum Komplettpreis von

Fast alle Programme mit deutschen Anleitungen nur **39,- DM**

BÜROPACK professionell

Art.-Nr. PDA019

Wer sagt denn, mit dem Amiga könne man nur spielen? Dieses Paket zeigt deutlich, daß Sie Ihren Amiga selbstverständlich auch professionell im Büro nutzen können! Büropack professionell ist eine Softwaresammlung besonderer Art, die für jeden Kaufmann oder Privatanwender interessant sein sollte, der nicht gleich tausende von D-Mark für ein Personal-Computer-System mit sündhaft teurer Software ausgeben will! Das Paket enthält folgende Programme: Oase 101 **Fibu deluxe+**. Dieses Programm erledigt Ihre Buchhaltung und schreibt Ihre Rechnungen! Oase 109 **Steuer 1991**. Jetzt können Sie Ihre Steuererklärungen schnell und einfach selber erledigen (mit preiswertem Update-Service für spätere Versionen); Oase 105 **Superdate deluxe**. Universell einsetzbare Dateiverwaltung, die sehr einfach zu bedienen ist; **Personal Write**. Eine überaus professionelle Textverarbeitung, die dennoch sehr einfach zu bedienen ist **(Test Amiga Special: sehr gut!)**. Außerdem enthält die Sammlung noch drei kleine ausgesprochen gute PD-Programme: Oase 26 **Giroman** verwaltet Ihre Girokonten; Oase 60 **Businesspaint** erstellt Präsentationsgrafiken von statistischen Werten (z. B. Umsatzerlöse usw.). Oase 63 **Tabellenkalkulation!**

Alle Programme sind ausführlich in deutsch beschrieben!

Der Komplettpreis für dieses Profipaket beträgt nur **199,- DM**

Übrigens: Wir sind OASE-Depot-Händler und haben alle OASE-Titel am Angebot!

SCHULPAKET

Das umfangreiche Lernpaket nicht nur für Schüler!

Chemie: **Elemente** das Periodensystem, **Moleküledatenbank** mit grafischer Darstellung, Mathematik: **R.O.M** umfangreiches Mathematikprogramm, **Mandelbrot** Apfelmännchengrafiken, Physik: **ABACUS** umfangreicher Elektronik-Grundlagenkurs, **Fields** elektrische Felder, Sprachen: **Perfect English** und **Latein** zwei Vokabeltrainer, Allgemein: **Schreibkurs** Maschinenschreiben, **Quizmaster** Abfragespiel mit Editor für eigene Fragen, **Stundenplan-Designer** Stundenpläne erstellen.

Alle Programme komplett deutsch

Komplettpreis für alle Programme nur **39,- DM**

Patrick Pawlowski
Software-Service
Kiefernweg 7, 2177 Wingst
Tel. 04777/8356 oder 04778/7294
BTX: *Pawlowski#

TOP 100

Die große deutsche Public-Domain-Sammlung

100 ausgewählte PD-Programme – die ideale Grundausstattung für jeden Amiga-500/1000/2000-Besitzer! Alle Programme sind in der Regel problemlos zu starten und haben, sofern eine Anleitung vorgesehen ist, **deutsche Beschreibungen!** Hier ein Auszug aus dem Komplettpaket: **Da Vinci** ein erstklassiges Malprogramm, **Business-Paint** Daten grafisch darstellen, **Geo** Erdkunde, **Analysis** Funktionen berechnen, **Video** Videoverwaltung, **AmiDat** Dateiverwaltung, **PowerPacker** Programme komprimieren, **Diskspeed** Geschwindigkeitstest, **Rechentrainer** Lernprogramm, **Boulder** kennen Sie Boulder Dask?, **Roll On** ein tolles Geschicklichkeits-Spiel, **Lucky Loser** Geldspielautomat, **Berserker** optimaler Virenkiller, **Dir Utility** vereinfacht den Umgang mit dem CLI, **Diskcat** katalogisieren Sie Ihre Disketten, **Disk-Label-Druck** Labels drucken, **Pit Dry Gen** erstellen Sie Ihren Druckertreiber, **Drip** das absolute Superspiel, **Maze Man** Pac Man-Spiel, **Noch Eins** ein tolles Breakout-Spiel, **Ahoi!** Schiffe versenken, **MS-Text** leistungsfähige Textverarbeitung, **Elements** das Periodensystem, **SD-Backup** Festplattensicherung, **Professionel D** eins der besten Kopierprogramme, **PCopy** ein weiteres Kopierprogramm, **GPrint** ein Grafikdruckprogramm, **Steinschlag** Tetris-Variante, **3D-Labyrinth**, **Exyptian Run** ein interessantes Actionspiel, **Icon Assembler** eigene Icons erstellen (bewegt), **Pointer-Animator** erstellen Sie einen bewegten Mauszeiger, **Fast Disk** optimiert Disketten, **Mastermind** das bekannte Spiel, **Chess** Schachspiel, **Boot Intro** Bootblock-Laufschrift, **WB-Pic** Bilder als Workbenchhintergrund, **Deluxe Hamburger** lustiges Ballerspiel, **Mega WB** Riesen-Workbench, **Sonix-Musik**, und **weitere 60 Programme** aus allen Bereichen!

NEU! *Jetzt aktualisierte Zusammenstellung*

100 Programme mit deutschen Anleitungen nur **99,– DM**

Bestellen Sie einfach mit dem Bestellcoupon oder formlos per Brief oder Postkarte. Sie können Ihre Bestellung selbstverständlich auch telefonisch oder per Fax aufgeben. Die Angebote sind freibleibend. Druckfehler und Irrtum sind vorbehalten. Die Lieferung erfolgt schnellstmöglich per Post.
Versandkosten: Vorauskasse (bar oder Scheck) DM 5,00, Nachnahme DM 8,00
Die Lieferung ins Ausland ist nur gegen Vorauskasse zzgl. DM 15,00 möglich!

Bestell-Coupon

Hiermit bestelle ich folgende Artikel aus Ihrem Sortiment: (bitte ankreuzen)

O TOP 100	O Musikpaket
O Superpack 50	O Schulpaket
O Superpack II plus	O Büropack
O Spielesammlung	O ______
O ______	
O ______	

O Ich bezahle per Vorauskasse
O Ich bezahle per Nachnahme

Außerdem erhalte ich die Einsteigerdiskette mit Anti-Virus-Programm sowie Ihren gedruckten Katalog!

Name: ______
Straße: ______
Ort: ______
Datum: ______ Unterschrift: ______

Test

3,5-Zoll-HD-Laufwerk

High Density für wenig Geld?

Mehr Daten auf der Disk – "HIDE" bringt es. Das HD-Laufwerk von FSE verspricht Kapazitäten bis zu 1,6 MByte.

HD-Disketten sind im PC-Bereich inzwischen zum Standard geworden. Auf dem Amiga war in dieser Richtung bisher kaum etwas zu hören. Das hat einen einfachen technischen Grund: Die Amiga-Diskettenhardware ist nicht in der Lage, die doppelte Datengeschwindigkeit von HD-Laufwerken zu verarbeiten, so daß der Anschluß eines handelsüblichen HD-Laufwerks an den Diskport nicht möglich ist.



Bild 1. High-Density-Laufwerk von FSE

Die Qualität des Magnetmaterials von Disketten wird in "Bits pro Inch" (bps) gemessen. Eine Normung der Disketten verlangt, daß diese Schreibdichte mit Kürzeln belegt wird. So bedeutet "DD" nichts anderes als "Double Density" (englisch: Doppelte Dichte). Dies entspricht den Standard-Amiga-Disks. HD ("High Density") arbeitet mit der doppelten Kapazität. Um trotzdem HD-Disketten am Amiga zu verwenden, muß also etwas mehr Aufwand getrieben werden. Drei Möglichkeiten stehen zur Wahl:

1. Man rüstet das Laufwerk mit einem eigenen Controller aus und bindet diese Konstruktion wie ein Festplattensystem ein. Das ist ziemlich aufwendig und teuer.

2. Man produziert einen speziellen Laufwerkstyp, der mit halber Drehgeschwindigkeit läuft. Damit ist die Datenrate wieder Amiga-kompatibel. Commodore bringt gerade ein solches Laufwerk auf den Markt; Amiga OS 2.0 unterstützt dies bereits.

3. Es wird getrickst: Benutzt wird ein normales HD-Laufwerk, allerdings nur im DD-Modus direkt am Disk-Controller. Statt der MFM-Modulation wird aber die effizientiere GCR-Codierung eingesetzt; die geringere Fehlersicherheit der Codierung wird von der erhöhten Fehlersicherheit des Laufwerks wenigstens teilweise aufgefangen.

HIDE

Nach Lösung 3. arbeitet "HIDE" von FSE. Das 295,– DM teure Paket enthält ein externes 3,5-Zoll-HD-Laufwerk und die nötige Treibersoftware.

Das Laufwerk, solide verarbeitet, mit durchgeschliffenem Bus, ist bis auf einen Umschalter für den Betriebsmodus auf der Rückseite von einer normalen externen Floppy nicht zu unterscheiden. Gefüttert werden kann es aber mit HD-ebenso wie mit DD-Disketten.

Die Treibersoftware aber hat es in sich. Bestehend aus dem "hide.device" und einigen Hilfsprogramm zum Anmelden und zur Fehlerabfrage, ermöglicht es den Parallelbetrieb des Laufwerks im HD- und im DD-Modus.

Dazu werden neben den üblichen Geräten DF0: bis DF3: für jedes HIDE-Laufwerk ein entsprechendes DF4: bis DF7: angemeldet. Ein HIDE-Laufwerk extern am Amiga 500 wäre dann unter DF1: als DD-und unter DF5: als HD-Laufwerk ansprechbar. Die eingelegte Diskette wird von den Treibern automatisch als DD oder HD erkannt und zugeordnet. Die "neuen" Geräte DF4: bis DF7: werden genau wie die alten angesprochen, inklusive Formatierung oder Zugriff mittels Disk-Monitor – nur eben mit mehr Platz.

Es existieren zwei Versionen des "hide.device": einmal eine schnellere Version mit 19 Sektoren pro Spur und eine zweite, langsamere mit 20 Sektoren pro Spur. Mit der 19-Sektoren-Version können Sie auf einer HD-Disk 1,56 MByte, mit der 20-Sektoren-Version 1,64 MByte unterbringen; geschrieben werden nicht 80, sondern 82 Spuren. Die 20-Sektoren-Version verlangt allerdings besseres Diskettenmaterial.

Das "hide.device" hat noch mehr Features aufzuweisen: So ist es im DD-Modus gerade auf Turbo-Amigas schneller als das "trackdisk.device", ermöglicht automatisches "NoClick" und hat einige Bugs weniger (wenigstens gegenüber der Kickstart-1.3-Version). Freundlicherweise liegt dem Paket ein Tool names "HideGrab" bei, mit dem es möglich ist, auch reine DD-Laufwerk über das "hide.device" anzusprechen und dessen Vorteile zu nutzen.

Geschwindigkeitsmäßig tut sich das Laufwerk im HD-Modus gegenüber einem normalen DD-Laufwerk nicht viel – die leichte Geschwindigkeitssteigerung ist auf das optimierte "hide.device" zurückzuführen. Durch Benutzung des Fast-File-Systems (auch unter 1.3 möglich) läßt sich aber zusätzlich die Geschwindigkeit steigern.

Im Betrieb erwiesen sich Software und Hardware als zuverlässig, obwohl die Datensicherheit aufgrund des technischen Verfahrens im HD-Modus als geringer einzuschätzen ist. Als reines "Arbeitstier" ist HIDE wenigstens im HD-Modus also nicht zu empfehlen – für Archivierungszwecke oder Backups ist es dagegen brauchbar.

Die Erklärung der Datenaufzeichnungsformate DD, HD, MFM und GCR finden Sie in unserem Computerlexikon. ❏

Jürgen Stephan (ow/tb)

FSE 3,5"-HD-Laufw.

AMIGA DOS

Testurteil 3'92

gut	**82%**
Anleitung:	85%
Installation:	80%
Bedienung:	–
Tauglichkeit:	80%
Preis/Leistung:	83%

Testmuster: FSE Frank Strauß Elektronik
Preis: 295,– DM
Konfiguration: alle Amiga
Einsatzbereich: universell

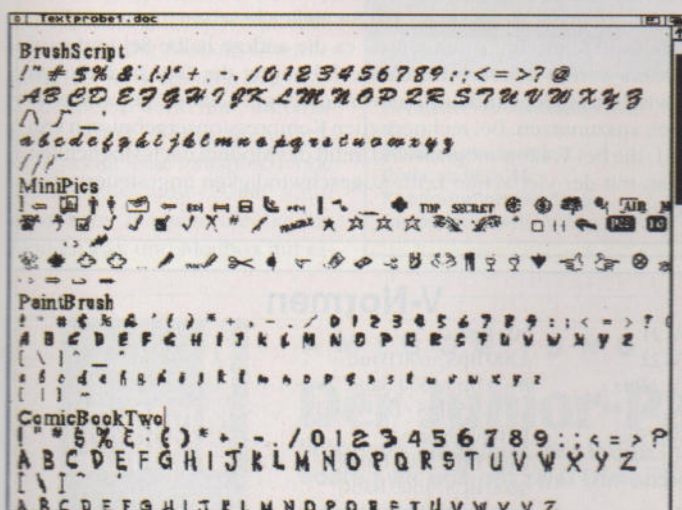
Outline-Fonts

Neue Post-script-Schriften

Zusammen mit der Erscheinung des DTP-Programms »Publishing Partner Master« wurden von SoftLogik auch hochwertige Bildschirm- und Druckerzeichensätze vorgestellt.

Die beim Amiga im Lieferumfang enthaltenen Zeichensätze liegen im Bitmap-Format vor. Sie zeichnen sich durch schnelle Verfügbarkeit aus und nehmen relativ wenig Speicherplatz ein. Ein Nachteil ist, daß für jede Bildschirmauflösung und Schriftgröße zusätzlich eine eigene Datei vorliegen muß. Das wahre Manko der Bitmap-Zeichensätze offenbart sich jedoch erst, wenn der Text skaliert (vergrößert oder verkleinert) oder ausgedruckt wird. Die Skalieroperation vergrößert nicht wie erwartet den Buchstaben als Objekt, sondern die Punkte, aus denen er sich zusammensetzt (Bild), so daß beispielsweise Linien unproportional dicker werden.

Dem Auflösungsvermögen des Druckers wird nur in etwa Rechnung getragen, denn die ursprünglichen Amiga-Daten müssen wegen wesentlich kleinerer Punktgrößen des Druckers vergrößert werden (häufig um Faktor drei). Danach erst hat eine Bitmap-Struktur das erwünschte, seitenfüllende Format. Das Ergebnis solcher Anpassungsversuche ist meist nicht zufriedenstellend, es sei denn, daß besonders ausgefeilte Skalieralgorithmen verwendet werden (mit Antialiasing-Routinen, die zwischen die Treppchen entsprechende Zwischenstufen einfügen). Diese Nachteile entstehen bei sogenannten Hüllkurven-Schriften (auch Outline-Fonts genannt) nicht.



Ausgefallene Schriften für verschiedene Zwecke

Zwar kennt die Welt der Typografie zahlreiche Formate für Hüllkurven-Schriften und beinahe jeder Schriftenhersteller schwört auf sein eigenes (Adobe hat das "Type 1-" und "Type 3-Format" entwickelt, Apple setzt auf das Konkurrenzprodukt "TrueType"). Allen ist aber gemeinsam, daß die Buchstaben durch Umrißlinien definiert sind, die mathematisch als Vektoren bezeichnet werden. Mit gleichem Prinzip arbeiten auch Vektorzeichenprogramme, mit denen in der Regel solche Schriftsätze entworfen werden – Beispiel: »FontDesigner«, siehe AMIGA DOS 2/92, Seite 38).

Das Datenformat legt im Gegensatz zu den Bitmap-Zeichensätzen keine Schriftgröße fest und ist deshalb von der Auflösung der Ausgabegeräte unabhängig. Bei der Darstellung solcher Zeichensätze wird durch vorherige Berechnung die vorhandene Auflösung des aktuellen Ausgabegeräts berücksichtigt, und dann erst werden die entsprechenden Bitmaps erzeugt (beispielsweise 70 dpi für Monitor oder 360 dpi für Drucker). So wird jedes Gerät optimal genutzt, und es spielt keine Rolle mehr, unter welcher Auflösung eine Schrift verwendet wurde. Die gute Qualität muß jedoch in der Regel durch verlängerte Rechenzeiten beim Druck bezahlt werden.

Die vorliegenden Schriften der Postscript-Typothek liegen im "Type 1 Postscript-Format" vor (von der Firma Adobe entwickelt). Die Schrift wird durch zwei Dateien definiert, wobei nach Übereinkunft der Schriftname (zum Beispiel Times) von einem sogenannten "tag" (sprich: täg) erweitert wird und die zwei Dateien unterscheidet. "Times.afm" bezeichnet dabei die "adobe font metrics", hier sind Angaben zu Laufweiten und proportionalen Abständen festgelegt (Kerning). "Times .pfb" hingegen enthält codiert und im Binärformat die eigentliche Schriftdefinition. Das Codierformat erlaubt dem Anwender keine Modifikationen an der Datei, so daß die Urheberrechte des Herstellers weitestgehend gewahrt werden. Zusätzlich zu diesen beiden sind bereits berechnete Bildschirmfonts vorhanden (von 10 bis 24 Punkt).

Bislang liegen vier Font-Pakete mit insgesamt 28 Schriftfamilien für verschiedene Schwerpunkte vor (zum Beispiel: für Design oder Zeitungsgestaltung). Die Installation ist denkbar einfach: Man kopiere die entsprechende Schublade in das von Publishing Partner verwendete Schriftverzeichnis. Mit dem Font-Manager muß lediglich der entsprechende Eintrag in den Voreinstellungen vermerkt werden, dann sind die Schriften verwendbar.

Entsprechend der Konzeption der Outline-Fonts, benötigen diese ungleich mehr Speicherplatz als Bitmap-Fonts, was aber der einzige gravierende Nachteil ist. Falls viele Zeichensätze benutzt werden sollen, muß man schon eine Festplatte angeschlossen haben, denn die Dateien brauchen durchschnittlich 50 KByte.

Diesem Paket liegt kein Konvertierprogramm für die Herstellung von Amiga-Bitmap-Fonts bei, der Einsatz ist also auf den Publishing Partner beschränkt. Die Qualität der Schriften ist wie erwartet sehr gut, was auch bei der Bildschirmdarstellung und beim Druck mit Punktmatrixgeräten deutlich auffällt. Die Schriften können jedem empfohlen werden, der seine Ausstattung komplettieren möchte. ❑

Dusan Zivadinovic (vb)

Outline-Fonts

AMIGA DOS

Testurteil 3'92

gut	**88%**
Anleitung:	–
Installation:	90%
Bedienung:	80%
Tauglichkeit:	90%
Preis/Leistung:	90%

Hersteller: DTM
Testmuster: Hersteller
Preis: Newsletter ca. 160 DM, Starter Pack ca. 160 DM, Classic Fonts ca. 350 DM, Designer Fonts ca. 350 DM
Konfiguration: für alle Amiga
Einsatzbereich: universell

DFÜ-Hardware

Modems für jeden Geschmack

Seit einigen Jahren boomt die Telekommunikation. Modems werden immer billiger, die Mailbox-Szene kann wöchentlich mit neuen Boxen aufwarten. So ist es nicht verwunderlich, wenn es immer neue Modems gibt.

Im Rahmen dieser "Modem-Flut" haben unsere Redaktion auch einige neuere Geräte erreicht. Das erste ist ein postalisches 2400-Baud-Modem, "Dataphon M 4805 MNP" der Firma Wörlein. Im Gegensatz zu den anderen beiden Geräten, ist der Betrieb dieses Modems am Telefonnetz der deutschen Bundespost erlaubt. Dies äußert sich unter anderem darin, daß verschiedene Einschränkungen und Modifikationen gegenüber einer nicht-postalischen Version vorgenommen wurden.
Das Dataphon-Modem stellt die Übertragungsmodi V.21, V.22, V.22bis und V.23 zur Verfügung. Zur Datensicherung kann MNP bis Klasse 5 sowie V.42bis verwendet werden. Neben dem Quasi-AT-Standard werden zusätzlich noch die Befehle nach V.25bis unterstützt.
Im Test erwies sich das Modem als relativ zuverlässig. Verbindungen von 300 Baud bis hin zu 2400 BpS mit V.42bis wurden ohne Fehler aufgebaut, auch wenn das Modem auf der Gegenseite nicht den Postaufkleber hatte. Lediglich im BTX-Modus kam es zu einigen unerklärlichen Abbrüchen, was jedoch auch an der Nebenstellenanlage liegen könnte.
Negativ fiel nur wieder auf, welche extremen Preiszuschläge der Aufkleber mit der ZZF-Nummer verursacht; das Modem schlägt mit knapp 900,– DM recht heftig auf die Geldbörse.
Bei dem zweiten Modem, dem "Discovery 2400P" von Datatronics, handelt es sich um ein wahrhaft kleines Gerät. Mit 12,5x6,5x3 cm paßt es in jede Tasche oder in jeden Koffer. Dieses Modem ermöglicht Übertragungen nach V.21, V.22 und V.22bis. MNP oder V42 sind nicht unterstützt. Im Betrieb erweist es sich als zuverlässig, jedoch kann man bei Verwendung an größeren Rechnern meist nicht sehen, was das Modem mit seinen LEDs anzeigt, da diese meist erfolgreich vom Gehäuse des Rechners verdeckt werden.
Das Handbuch enthält eine Reihe interessanter zusätzlicher Informationen (Steckerbelegungen von verschiedenen Telefonbuchsen, Signal-Infos), die man leider bei anderen Herstellern vermißt. Mit einem Preis von 220,– DM ist es durchaus erschwinglich.
Das schnellste Modem im Bunde ist ein echter Newcomer. Mit seiner Leistung ist es ein Konkurrent für die bisherigen Marktführer US Robotics und die Telebit-Modems.
Im Test hatten wir die einfachere Ausführung des ZyXEL-Modems, es ist außerdem auch eine Version erhältlich, bei der die Einstellung über Tastatur und LCD-Display vornehmbar ist. Die ZyXEL-Modems verfügen serienmäßig über den V.32bis-Standard, somit kann es mit den gängigen High-Speed-Modems in der Regel mit mindestens 9600 BpS kommunizieren. Selbstverständlich verfügt es auch über V.42bis.

Warum ist das Modem schneller als die Leitung?

Nahezu alle Modems, die über MNP5 oder V.42(bis) verfügen, erlauben es, mit mehr als der auf der Leitung übertragenen BpS-Zahl angesteuert zu werden. Bei "normalen" Modems wird auf der Leitung nur der "Nennwert" übertragen, also beispielsweise 2400 BpS. Können die Daten dagegen komprimiert werden, so kann es sein, daß das Modem die Menge von 2400 Bit vielleicht schon in einer halben Sekunde überträgt, somit würde es die andere halbe Sekunde auf Daten warten müssen. Durch die Erhöhung der Geschwindigkeit zwischen Modem und Rechner wird es möglich, diese "gewonnene" Zeit auszunutzen. Bei rechnerischen Kompressionsergebnissen von 4:1, die bei V.42bis möglich sind, muß das Modem nach Möglichkeit also mit der vierfachen Leitungsgeschwindigkeit angesteuert werden.

Die AT-Befehle

"AT" ist eine Bit-Kombination, die es ermöglicht, daß das Modem sowohl die Geschwindigkeit der seriellen Schnittstelle als auch deren Einstellung (Start-, Stop-, Datenbits und Parität) feststellen kann. Danach folgen ein oder mehrere Befehle, die vom Modem ausgeführt werden.
Die Befehle, die danach direkt mit einem Buchstaben beginnen, sind inzwischen fast durchgängig herstellerunabhängig belegt, so daß man mit jedem Modem, das die AT-Befehle kennt, mit einigen Befehlen alle grundlegenden Aktionen durchführen kann.

V.25bis-Befehle

Die Befehle nach der V.25bis-Festlegung werden zur Zeit fast nur bei postzugelassenen Modems eingesetzt. Sie bestehen aus 3-Zeichen-Kombinationen, die eine Abkürzung des eigentlichen Befehls darstellen (zum Beispiel "CRN": Call Request with Number provided; Anruf der angegebenen Nummer (= ATD), "CIC": Connect Incoming Call; Anruf annehmen (A = ATA)).

V-Normen

V.21	300 BpS
V.22	1200 BpS / 600 Baud
V.22bis	2400 BpS / 600 Baud
V.23	1200/75 BpS / 600 Baud
V.25bis	Befehls-Norm
V.27ter	2400 BpS/1200 Baud
	4800 BpS / 1600 Baud
V.29	4800-9600 BpS / 2400 Baud
V.32	9600 BpS / 2400 Baud
V.32bis	7200-14400 BpS / 2400 Baud
V.33	12000-14400 BpS / 2400 Baud 4-Draht
V.42	Fehlerkorrektur-Protokoll
V.42bis	Fehlerkorrektur und Datenkompression

Begriffe in der DFÜ

BpS: Bit pro Sekunde, Anzahl von Ja-Nein-Informationen, die pro Sekunde übertragen werden.
Baud: Anzahl von Informationseinheiten pro Sekunde. Pro Informationseinheit können mehrere Bit übertragen werden.
Western-Stecker: Telefonsteckernorm, die praktisch überall außer in Deutschland verwendet wird; vier- oder sechsadrig.
TAE-Stecker: Deutsche Steckernorm zum Anschluß von Telefonen, Anrufbeantwortern, Modems etc.

Im Gegensatz zum direkten Konkurrenten US Robotics, kann das ZyXEL mit einer Geschwindigkeit von 57600 BpS angesteuert werden, was bei starker Kompression die einzige Möglichkeit ist, das Modem gut auszulasten. Dies wird vor allem dann wichtig, wenn der angekündigte 19200-BpS-Modus verfügbar wird. Bei einer maximalen Kompression von zirka 3:1 bis 4:1 sind so hohe Übertragungsraten natürlich nötig. Leider sind die aktuellen Amigas aber nicht in der Lage, so hohe Geschwindigkeiten zu benutzen. Abhilfe kann hier nur eine Karte mit seriellem Port leisten, der in der Lage ist, diese Geschwindigkeiten zu nutzen.

Als interessante Neuerung auf dem Modem-Markt haben sich ja schon seit einer Weile verschiedene Fax-Modems hervorgetan, die jedoch allesamt einen Nachteil haben – entweder sie stehen auf Betriebsart "Modem", dann ist es nicht möglich, einen Fax-Anruf entgegenzunehmen, oder sie stehen auf Fax, wobei entsprechend die Modem-Übertragung zu kurz kommt. Das ZyXEL-Modem erkennt automatisch, womit es angerufen wird, und bringt eine entsprechende Connect-Meldung. Somit kann man zum Beispiel die beiden Medien Fax und Mailbox auf einem Anschluß verwenden. Das englischsprachige Handbuch erklärt alle zur Verfügung stehenden Befehle und Register gut.

Im Test mit verschiedenen Modems auf der Gegenseite erwies sich das Modem als zuverlässig. Verbindungen mit Modems wie "Telebit T2500", "USR Dual Standard" oder auch nur einfacheren 2400-BpS-Modems wurden problemlos aufgebaut. Dabei erwies sich auch die Skepsis, ob denn nun der V.32bis-Standard wirklich auch ein Standard ist und bei verschiedenen Herstellern funktioniert, als unangebracht. Die 14400-BpS-vollduplex-Verbindung zu zwei verschiedenen Modems wurde einwandfrei gehalten, die Übertragungsraten entsprachen denen zwischen zwei HST-DS-Modems. Weiterhin wurde das Modem auch im Rahmen der Redaktions-Mailbox eingesetzt und hat in dieser Funktion etwa ein bis zwei Dutzend MByte übertragen. Dabei ist erfreulich anzumerken, daß sich das Modem auch im Dauerbetrieb kaum erhitzt.

Mit dem Leistungsumfang und der gezeigten Qualität kann sich der Newcomer ZyXEL mit Preisen von zirka 1500 DM beziehungsweise 1950 DM durchaus mit den alten Hasen "HST" und "Trailblazer" messen. Vor allem durch die – für "HST" schon lange angekündigte, aber bisher nicht verfügbar – Fax-Funktion bietet es doch einen recht guten Kaufanreiz. ❑

Garry Glendown (jb)

Dataphon M 4805 MNP

AMIGA DOS Testurteil 3'92

befriedigend	72%
Anleitung:	80%
Installation:	–
Bedienung:	80%
Tauglichkeit:	90%
Preis/Leistung:	65%

Hersteller: Wörlein GmbH & Co. KG
Muster von: Hersteller
Preis: ca. 900 DM
Konfiguration: alle Amiga
Besonderheiten: Postzulassung

ZyXEL U-1496

AMIGA DOS Testurteil 3'92

gut	85%
Anleitung:	85%
Installation:	–
Bedienung:	80%
Tauglichkeit:	95%
Preis/Leistung:	85%

Hersteller: ZyXEL Communications Corporation
Muster von: Elmar Elsner
Preis: ca. 1500 DM
Konfiguration: alle Amiga
Besonderheiten: keine Postzulassung

Discovery 2400 P

AMIGA DOS Testurteil 3'92

gut	80%
Anleitung:	85%
Installation:	–
Bedienung:	80%
Tauglichkeit:	80%
Preis/Leistung:	90%

Hersteller: Datatronics
Muster von: Fischer Hard- und Software
Preis: ca. 220 DM
Konfiguration: alle Amiga
Besonderheiten: keine Postzulassung

Datenpacker

PowerPacker 4.0

Relativ unbemerkt von den Massen, ist auf der Expo in Köln auch eine neue Version des bekannten Datenpackers »PowerPacker« erschienen.



Jetzt mit ARexx-Port: »PowerPacker 4.0«

Die Version 4.0 ist nun wirklich nicht die erste Release des »PowerPacker«, und vielen Amiga-Benutzern dürfte auch die ältere Shareware-Version 2.3b (auf Fish 253) bekannt sein. Stellt sich automatisch zuerst die Frage: Was hat sich geändert?

Am auffälligsten ist mit Sicherheit die zusätzliche Version, die sich ab sofort auf der Diskette befindet. Sie benötigt mindestens Kickstart 2.0, was bereits darauf hinweist, daß einige der wirklich interessanten Neuerungen der aktuellen Betriebssystemversion unterstützt werden. So öffnet der »PowerPacker« zum Beispiel einen sogenannten "Public Screen", ein 2.0-Feature, das es ermöglicht, programmfremde Fenster (zum Beispiel eine Shell) auf diesem Screen zu öffnen. Ebenso können nun der Bildschirmmodus oder der Zeichensatz frei eingestellt werden, und die Iconify-Funktion (früher "Sleep") verwendet die standardisierte Methode des Application-Icons. Zudem ist die 2.0-Version, obwohl sie mehr Funktionen hat, auch noch kürzer als der "1.3-Kollege", wenn auch nur geringfügig.

Die zweite große Neuerung, die dem Programm zuteil wurde, ist der ARexx-Port. Dem Anwender stehen alle wichtigen Funktionen des Programms extern zur Verfügung. Zusätzlich gibt es auch noch ein paar Befehle, die nur für ARexx-Scripts (Programme, die unter ARexx gestartet werden) von Bedeutung sind und somit von der Intuition-Oberfläche (Workbench) aus nicht aufgerufen werden können. Ein Beispiel-Script, das die Möglichkeiten, die Sie mit dem ARexx-Port besitzen, ein wenig aufzeigt, finden Sie übrigens in dieser Ausgabe im Tips-&-Tricks-Teil.

Während 1.3-Anwender jedesmal, wenn Sie ein ARexx-Script starten möchten, den Namen und eventuelle Parameter in ein String-Gadget (Eingabefeld für Text) tippen müssen, existiert unter 2.0 ein zusätzliches Menü, welches Sie mit zehn beliebigen ARexx-Programmen vorbelegen können. Diese Einstellungen können selbstverständlich auch abgespeichert werden. Insgesamt ist die Einbindung des ARexx-Ports als gelungen zu bezeichnen, was man bei anderen Programmen leider nicht immer behaupten kann. Lediglich ein kleiner, aber folgenschwerer Fehler, der das Kommando "Save" auf 68000-Rechnern zum Absturz brachte, hätte den Eindruck getrübt, wäre er nicht inzwischen vom Autor beseitigt worden. Besitzer einer solchen Version können aber nach Aussagen des Autors bei UGA gegen Einsendung der Originaldisk plus Rückporto die neue Version erhalten. Dem Autor dieses Artikels liegt eine funktionierende Version vor, die am 6.12.91 kompiliert wurde.

Soviel zu den nach außen sichtbaren Änderungen, denn auch intern hat sich etwas getan. Das Wichtigste: Der »PowerPacker« ist jetzt etwas effizienter und hat somit in diesem Punkt wieder mit dem Hauptkonkurrenten »Imploder 4.0« gleichgezogen. Im Vergleich gewinnt, wie der Tabelle zu entnehmen, mal der eine, mal der andere Mitstreiter. In Sachen Geschwindigkeit hingegen liegt der »PowerPacker« eindeutig vorne. Ebenso wie der »Imploder« hat der »PowerPacker« die Möglichkeit, den Decrunch-Code in eine Library (Sammlung von Programmierbausteinen) auszulagern.

Wenn man nur die möglichen Packergebnisse vergleicht, tun sich »PowerPacker« und »Imploder« nicht viel. Die Stärke des »PowerPacker« liegt in der bequemeren Bedienung, wobei insbesondere der ARexx-Port eine nicht unmaßgebliche Rolle spielt. Der »Imploder« dagegen ist Freeware und somit kostenfrei nutzbar. ❑

Christoph Teuber (vb)

		PowerPacker 4.0		Imploder 4.0	
Testfile	Länge	Länge	Zeit	Länge	Zeit
Addmenu	10192	3672	0:03 min	3968	0:07 min
Filemaster	73160	30004	0:55 min	29880	1:49 min
SuperDuper2	45272	25780	0:17 min	25752	0:23 min
Omega	547940	274192	6:33 min	275364	10:18 min

Testbedingungen: Getestet wurde auf einem A2000 mit 68000er Prozessor und 5 MByte Speicher. Der Speedup-Buffer des »PowerPacker« stand auf "Large", ebenso lief der »Imploder« im Turbo-Mode. Bei beiden Programmen war die Library-Option eingeschaltet und die effizienteste Packstufe aktiv.

Vergleich zwischen »PowerPacker 4.0« und »Imploder«

Glossar

ARexx-Port: ARexx ist eine Programmier- oder genauer gesagt eine Script-Sprache, die bei den Amiga-Modellen A500 Plus und A3000 im Lieferumfang enthalten ist. Mit dieser Sprache ist es möglich, aus einem Programm heraus andere Programme "fernzusteuern". Man muß das aktuelle Programm nicht verlassen, um mit einem anderen Programm in Kontakt zu treten – beispielsweise um Daten auszutauschen. Jede neue Software wird dementsprechend über einen sogenannten ARexx-Port verfügen, der letztendlich die Schnittstelle zwischen ARexx und den einzelnen Programmen darstellt.

PowerPacker

AMIGA DOS

Testurteil 3'92

gut	80%
Anleitung (D) Disk:	60%
Installation:	70%
Bedienung:	90%
Tauglichkeit:	90%
Preis/Leistung:	80%

Hersteller: UGA
Testmuster: UGA
Preis: 39,- DM
Konfiguration: alle Amigas
Einsatzbereich: semiprofessionell

Audio Master IV

Des Maestros neue Kleider

Es hat ja nicht lange gedauert, bis aus einer "III" eine "IV" wurde. Die Rede ist von der neuesten Version des »Audio Master«, dem Profi unter den Audio-Digitalisier-Programmen.

Audio Master IV« kann jetzt das Tonhöhen-Zeit-Verhältnis (pitch/time) besser kontrollieren. Dies wird benötigt, um das Verhältnis der Abspielgeschwindigkeit zur Tonhöhe des Samples besser zu variieren. Die Tonhöhe kann dabei in Echtzeit ("realtime") verändert werden.

Was ist neu?

Ebenfalls neu ist der digitale Filter, der durch die drei Funktionen "Boost" (steht für Anhebung), "Cut" (schneiden) und "Pass" (Nutzung der Wellenform oberhalb der hohen Frequenzen und unterhalb der niedrigen Frequenzen; entspricht ungefähr einem Entfernen der mittleren Frequenzen) präsentiert wird. Dieser Filter erlaubt es, störende Frequenzspitzen oder schwache Signalstellen mit den drei genannten Funktionen zu bearbeiten.

Des weiteren stehen "Zoom" und "Scroll" jetzt ebenfalls in Echtzeit zur Verfügung. Der Hardware-Audio-Filter, repräsentiert durch Heller- oder Dunklerwerden der Power-LED, wird automatisch vom Programm kontrolliert. Die DC-Offset-Kompensation zum Ausgleichen der Trägergleichspannung kann vom Benutzer derweil beeinflußt werden.

Weitere wichtige neue Features des »Audio Master IV«:

– Marker für Zeit und Position sind direkt ablesbar.

– Verbesserte Echo-, Mix- und Ramp-Funktionen ("Ramp" bedeutet ein An- oder Absteigen des Tones oder des Sounds, je nachdem, ob man "down-" oder "up-rampt").

– »Audio Master IV« enthält einen Perfect-Sound-Support. Dies gilt für diejenigen, die einen Perfect-Sound-Sampler besitzen und diesen zusammen mit »Audio Master IV« betreiben wollen. »Audio Master IV« meldet sich im Gegensatz zu seinen Vorgängern ziemlich rasant auf dem Bildschirmen. Das Programm arbeitet, wie seine Vorgänger, mit dem gesamten erreichbaren Speicher, benutzt diesmal aber ein Meßraster zum Positionieren der Marker. Zusammen mit den beiden Samplern »Perfect Sound« und »Sound Master«, der ja bekanntlich schon mit »Audio Master III« als Paket ausgeliefert wurde, ergeben sich wie gewohnt hervorragende Ergebnisse. Die neu hinzugekommenen Funktionen bieten somit eine weitere Verbesserung eines ohnehin schon hochwertigen Produktes.



Bild 1. »Audio Disk Jockey« – Bonusprogramm für Sample-Bänke

Zusätzlicher Bonus

Neu hinzugekommen ist allerdings auch ein neues Bonusprogramm: eine Sample-Bank-Verwaltung namens »Audio Disk Jockey«, bei der es sich um einen Sample-Player handelt und deren Zusammenstellung einzig dem Benutzer unterliegt. Die Bänke können abgespeichert und geladen werden. Bis zu 180 Sounds können somit auf einfachste Weise zu Jingles zusammengestellt werden, was jedem Rundfunkmoderator zur Ehre gereichen wird. Die einzelnen Samples lassen sich hier im sogenannten "Edit-Mode" nochmals beeinflussen; ebenso läßt sich der Hardware-Filter des Amiga an- oder abschalten.

Wer »Audio Master« bisher noch nicht kannte, wer aber für seine Audio-Digitalisierung das Beste will, der kommt am neuen »Audio Master IV« nicht vorbei. Wer bereits »Audio Master III« besitzt, der findet bei »Audio Master IV« die eben genannten neuen Funktionen. Ob er sie braucht, steht auf einem anderen Blatt. »Audio Master IV« ist auf jeden Fall das Programm, um vernünftige Samples zu erstellen. ❑

(jb)

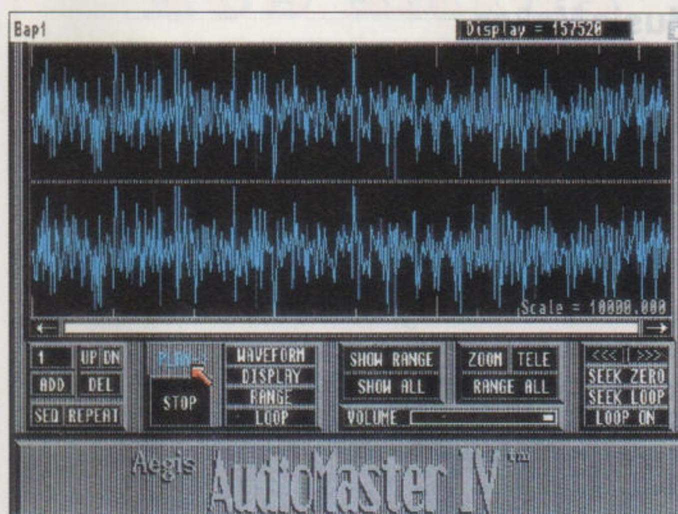


Bild 2. »Audio Master IV« – Standard für Audio-Digitizer

Audio Master IV

AMIGA DOS

Testurteil 3'92

sehr gut 93%

Anleitung (englisch):	80%
Installation:	70%
Bedienung:	90%
Tauglichkeit:	95%
Preis/Leistung:	95%

Hersteller: Oxxi inc.
Testmuster: ESD
Preis: 198,– DM
Konfiguration: alle Amiga

Test

Music Maker

Acht Stimmen für ein Halleluja

"Willkommen im Reigen der Sound-Programme ..." Da dieses Programmgenre recht ausführlich vertreten ist, muß man im Falle von »Music Maker« schon ein wenig näher hinsehen, um die Frage "Komplettes Musikstudio oder nur ein weiterer Sound-Editor?" zu klären.

Prinzipiell gibt es zwei Arten von Musik-Editoren: notenorientierte und zeichenorientierte. Bei notenorientierten Programmen (wie »Sonix« und »Deluxe Music Construction Set«) werden die für Musikstücke notwendigen Angaben wie Tonhöhe, Tonlänge und Lautstärke in der konventionellen Notenschreibweise eingegeben. (Für diejenigen, die sich im Musikunterricht nur mit Papierflugzeugen und/oder Leidensgenossen oder -genossinnen beschäftigt haben: "konventionell" bedeutet fünf waagerechte Linien, auf denen sich leere und gefüllte kleine Kreise befinden.)

Die zeichenorientierten Programme (wie »Face the Music« und die unzähligen SoundTracker-Clones) hingegen erwarten die Eingaben in Form von Zahlen und Buchstaben. Das erspart dem Benutzer erstens die (wenn auch geringe) Mühe des Notenlernens und erlaubt ihm zweitens schnellere Eingaben über die Tastatur. Der Überblick über einen Melodieverlauf bleibt bei diesem Verfahren allerdings leicht auf der Strecke.

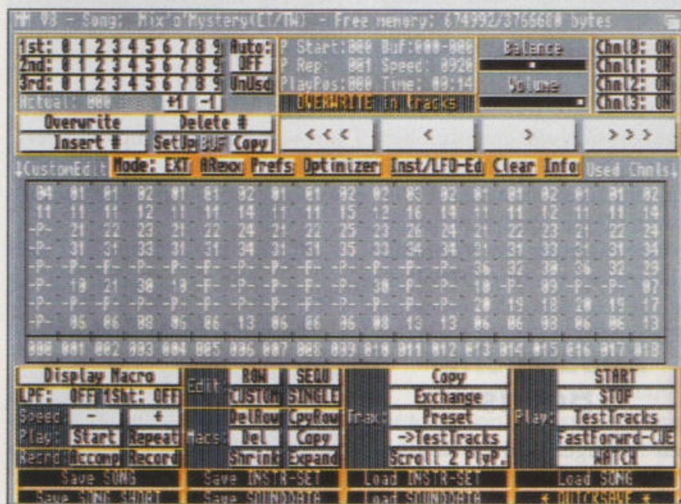


Bild 1. Leistungsfähig und unübersichtlich: der Track-Editor ...

Auf dem Amiga überwiegen die zeichenorientierten Editoren, da anscheinend die meisten musikinteressierten Computer-Benutzer eher Programmierkenntnisse haben als musikalisches Wissen.

Thomas Winischhofers »Music Maker« ist ein neuer Vertreter dieser zeichenorientierten Editoren. Wie bei den meisten derartigen Programmen, müssen Melodien und Begleitungen taktweise eingegeben werden. Diese Takte (hier: Makros) werden dann zum kompletten Musikstück (hier: Track) zusammengesetzt.

»SoundTracker« und Konsorten verführten den Benutzer dazu, einmal programmierte Takte möglichst oft zu verwenden. Dies spart zwar Speicherplatz, führt im Endeffekt aber zu nicht sehr abwechslungsreichen Musikstücken. »Face the Music« hingegen besann sich wieder auf das Konzept des »Deluxe Music Construction Set«: Stücke werden dort nicht aus Blöcken zusammengesetzt, sondern fortlaufend notiert, wobei Bearbeitungsfunktionen, wie Kopieren und Einfügen, dennoch für Bedienungskomfort sorgen.

»Music Maker« geht einen – durchaus sinnvollen – Mittelweg. Musikstücke bestehen, wie erwähnt, aus Blöcken. Wenn der Benutzer jedoch einen bereits eingegebenen Block noch einmal benötigt, verwendet er nicht denselben Block ein zweites Mal, sondern kopiert dessen Inhalt in einen weiteren Block. Dadurch bleibt das Stück übersichtlicher, und es ist möglich, kleinere oder größere Veränderungen im zweiten Block vorzunehmen, was sehr zum Abwechslungsreichtum eines Stückes beiträgt. Erst wenn die Komposition fertig ist, wird sie vom Programm automatisch optimiert, das heißt, doppelte oder gar ungenutzte Blöcke werden entfernt, um Speicherplatz zu sparen.

Die Notenwerte (verwendetes Instrument, Tonhöhe, Tonlänge, Lautstärke) kann der Anwender im »MacEd«, dem Makro-Editor, sowohl über die Tastatur als auch mit Hilfe der Maus eingeben. In die Melodien lassen sich bereits eine ganze Reihe von Effekten integrieren, beispielsweise Modulieren der Lautstärke oder Tonhöhe eines bereits angespielten Tones, Portamento (Übergehen eines Tones in einen anderen) oder Loudness (logarithmisches Reduzieren der hohen Töne; ein Abfallen, das nicht geradlinig erfolgt).

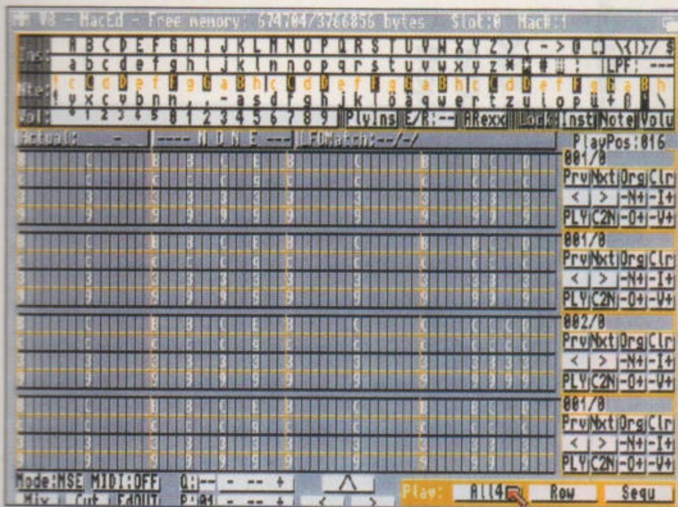
Wer etwas weniger mathematisch an das Komponieren von Musikstücken herangehen möchte, der kann Melodien in Echtzeit einspielen – und zwar entweder über die normale Amiga-Tastatur oder aber über ein angeschlossenes MIDI-Keyboard. Eine Quantisierungsfunktion hilft dabei, kleinere Spielungenauigkeiten auszumerzen.

Trickreiche acht Stimmen

Dem Amiga stehen Hardware-bedingt "nur" vier Tonkanäle zur Verfügung. Der »Music Maker« hebt – wie seine Konkurrenten »OctaMED«, »Oktalyzer« und »Face the Music« – diese Beschränkung auf: Er bietet dem eifrigen Komponisten bis zu acht einzelne Stimmen, was auch für detailliertere Begleitungen ausreicht. Möglich wird dieser Stimmenreichtum durch das Zusammenmischen von jeweils zwei Stimmen. Die hierdurch entstehenden Qualitätsverluste des Klanges bleiben meist im akzeptablen Rahmen. In bestimmten Situationen (beispielsweise bei sehr hohen und tiefen Tönen auf einem Mischkanal) muß jedoch zu kleinen Tricks gegriffen werden, um störendes Rauschen zu vermeiden. Dies ist nicht weiter kompliziert, stellt jedoch eine nicht zu unterschätzende Hürde dar für Musiker, die einfach nur schnell und unkompliziert mit einem Computer musizieren möchten, ohne sich mit technischen Einzelheiten auseinandersetzen zu müssen.

Eine originelle und wirkungsvolle (aber auch recht aufwendige) Methode, um Stücke mit acht Stimmen ohne jeden Qualitätsverlust abzuspielen, besteht darin, zwei Amigas über die serielle Schnittstelle zu verbinden und jeden der Computer vier Stimmen spielen zu lassen.

Seinem Anspruch, ein "komplettes Musikstudio" zu sein,


Bild 2. ... und der Makro-Editor

wird der »Music Maker« schon eher dadurch gerecht, daß er ein integriertes Sample-Programm besitzt.

Gewöhnungs-bedürftig

Auch wenn längere Folgen von Funktionsaufrufen durch die »ARexx«-Schnittstelle bequem in Makros (die nichts mit den Takt-Makros des »Music Maker« zu tun haben) zusammengefaßt werden können, liegt der Schwachpunkt des Programms eindeutig in der Bedienungsoberfläche: Ihr Konzept ist zwar durchaus durchdacht, hält sich aber an keine Standards, sondern geht eigene Wege, was das Einarbeiten sehr erschwert. Zudem sind – bedingt duch das Fehlen von Pull-down-Menüs – die Bildschirme mit Gadgets so überladen, daß für die Darstellung der eigentlich zu bearbeitenden Daten nur noch recht wenig Platz übrigbleibt.

Im Lieferumfang enthaltene Abspielroutinen sowie eine Library ermöglichen es, sowohl mit Interpreter-Sprachen wie »AmigaBASIC« als auch mit »C« und Assembler Stücke des »Music Maker« in eigenen Programmen zu verwenden. In BASIC können allerdings nur vierstimmige Stücke abgespielt werden. Drei Konvertierungsprogramme erleichtern das Umwandeln von Stücken, die als Sound-Tracker-, Oktalyzer-oder IFF-SMUS-Dateien vorliegen.

Die 170seitige Anleitung ist zwar detailliert, schreckt aber durch ihre ungeschickte Struktur, teilweise unvollständige oder sinnlose Sätze sowie Layoutfehler eher ab, als daß sie den Einstieg in das Programm erleichtert.

»Music Maker« ist alles in allem ein leistungsfähiger Editor, der allerdings eher für Programmierer gedacht ist, die ihre Programme musikalisch untermalen möchten. ❑

Hartmut Schumacher (jb)

Music Maker

AMIGA DOS

Testurteil 3'92

befriedigend 68%

Anleitung (D):	50%
Installation:	70%
Bedienung:	60%
Tauglichkeit:	75%
Preis/Leistung:	70%

Hersteller: Data Becker
Testmuster: Data Becker
Preis: 99 DM
Konfiguration: alle Amiga-Modelle mit 1 MByte Speicher

Kleinanzeigen

Biete Software

Fred Fish -560 Stand 12.12.91
10 Stück bei Vorkasse 20,– DM
je weitere PD-Disk +1,50 DM
Bei Nachnahme zuzüglich 5,– DM
M. Berger, Augsburgerstr. 375,
7000 Stuttgart 60, T.: 07 11/33 24 71 **G**

TV-FILM-SPORT-DAMEN IN SEXPOSEN, INFO + 2 SERIEN=20,-V-SCHECK + AUSWEISKOPIE. BEI KAUF DM10,– GUTSCHRIFT. Galesic, PF500411, 7STGT.50 **G**

AMIGA PUBLIC DOMAIN
Versand aller gängigen PD-Serien
PD-Diskette 1,80 DM
HM Computing, Tel. 0 67 27/51 46 **G**

ALIEN GRAPHICS
digitalisiert in z.Zt. bestmöglicher Qualität Vorlagen bis DIN A3 AB **1DM**
Tel. 02 09/7 48 73 **G**

Amiga-Abos!!! Amiga-Abos!!!
Tolle Demos, umfangreiche Anwender & super Spiele für wenig Geld gibt es bei: Stefan Hüls,Stichwort: Amiga-Abo,Dinxperloer Str. 87, 4290 Bocholt. Infos gegen 2,– Rückporto

BAVARIAM, die einzig wahre PD-Serie, jetzt mit 320 Disketten. Gratiskatalog bei Friedrich Neuper, 8473 Pfreimd, Postfach 72 **G**

Biete Hardware

DIN-A3-PLOTTER
Kein Spielzeug! **BAUSATZ** kompl.mit Gehäuse und Interface nur **DM 349,–!**
FERTIGGERÄT nur **DM 449,–!**
Bauplan **DM 10,–!** Auflösung 0,1 mm, Geschwindigkeit 70 mm/sec. Kostenloses Info bei P. Haase, Dycker Str. 3, 4040 Neuss 22, Tel.: 0 21 31/8 43 40 **G**

VORTEX-HD 60MB! f. AMIGA 500/1000, VB 1100 DM, TEL.: 0 89/84 31 87

Verschiedenes

******AMIGA-BILDERDIENST******
Farbausdrucke in Fotoqualität auf Papier o. Folie m. Xerox-Tintenstrahl- o. Calcomp-Thermotransfer-Drucker DM 6,– (Papier/DIN A4). Infos m. Druckmuster, Tel. 02 51/6 22 14. CGD Dr. Buddemeier, ***Schlesienstr. 40, 4400 Münster*** **G**

COMPUTERCLUB HOTLINE 0 86 42/62 79

Wer hat Bock, angetörnte Spiele u. kl. Movies zu machen? Spaß, Programme + Scripts vorh. T. 0 66 76/81 85

Club

Warum noch kein Mitglied im CCM e.V.?? Der Club für alle AMIGA-Anwender! Club-Zeitung + weitere Infos geg. 2,40 DM Rückporto. Große PD-Bibliothek. Computer-Club-Minden e.V., 4907 Bad Oeynhausen, PF 100905. Der Computer-Club in NRW !!!

Viele Clubs bieten wenig, einige Clubs bieten sogar recht viel, doch ein Club bietet alles: **EXTREM - Der bessere Club.** Für Systeme Amiga, C 64, PC, Atari ST, Game Boy, Nintendo, Game Gear, Lynx, Sega Mega Drive, Sega Master System & CDTV. Hier einige Leistungen: 30-seitige Clubzeitschrift, Hotline (0 28 74/35 70 Karsten/tägl. von 15-21 Uhr), Jahreswettbewerb, Clubausweis usw. Info erhaltet Ihr gegen 2,– DM Rückporto unter: Stefan Hüls, Stichwort: Clubinfo, Dinxperloer Str. 87, 4290 Bocholt

AMIGA DOS -TIP – die aktuelle Produktanzeige

Ihre Hotline für AMIGA-DOS-TIP

Tel.: 089/4391087 bis -89

Fax: 089/4391080

AMIGA DOS -TIP – die aktuelle Produktanzeige

Die Leserbrief-Seiten sind ein Pool, der für alles und jeden offensteht. Wenn Sie möchten, dann meckern Sie, geben Anregungen, üben Kritik an der Redaktion, Firmen, Software und Hardware, dem Service einer Firma, oder schreiben Sie einfach, was Ihnen auf dem Herzen liegt.

DMV-Verlag Redaktion AMIGA DOS Leserservice Postfach 250 3440 Eschwege

Bitte schreiben Sie Ihre vollständige Adresse nicht nur auf den Umschlag, sondern auch in den Brief. Bei Fragen legen Sie am besten einen an Sie selbst adressierten und ausreichend frankierten Rückumschlag bei.

Leserbriefe

Fragen zum Disk-Monitor

Zum Artikel "Flache Scheiben", in dem auf die Interna des Amiga-Diskettenformats eingegangen wird, habe ich eine Anmerkung, eine Frage und eine Bitte:

a) Der Autor erläutert ausführlich, wie die Position des Rootblocks zu berechnen ist (Seite 66 unten und Folgeseite). Das geht aber bei intakten Disketten wesentlich einfacher: Das dritte Langwort des Bootblocks enthält die Blocknummer des Rootblocks.

b) Das zweite Langwort des Bootblocks enthält eine Checksumme. Soll auch sie wie die Data-Block-Checksumme addiert zu den anderen Langworten 0 oder aber -1 ergeben? (Beide Versionen sind mir schon zu Ohren gekommen.)

c) Da ich (und vermutlich viele andere interessierte Leser auch) der Sprache C leider nicht mächtig bin, möchte ich Sie bitten, den Hash-Algorithmus nochmal in "normaler" Sprache zu erklären oder ein äquivalentes Assembler-Listing zu veröffentlichen.

A. Eigenwillig, Meckenheim-Merl



Zu a): Leider ist dem nicht so. Dieser Block enthält zwar nach den Unterlagen die Nummer des Rootblocks, aber in der Regel findet man hier lediglich eine Null. Unter Umständen erzeugen manche "PD-Install-Programme" hier den korrekten Zeiger. Da jedoch der Aufwand, die Korrektheit dieser Blockreferenz zu prüfen, größer ist, als gleich die richtige Blocknummer zu berechnen, wurde er einfach ignoriert. Übrigens kümmert sich auch das DOS nicht um ihn.

Zu b): Die Prüfsumme für den Bootblock muß, genau wie bei den anderen Blöcken auch, 0 ergeben. Allerdings wird sie nicht nur über Block 0, sondern auch über Block 1 berechnet, da beide als eine Einheit behandelt werden.

Zu c): Da eine Assembler-Version wohl für noch mehr Leute nicht zu verstehen ist, ist eine kurze Beschreibung wohl günstiger: Ausgangswert für die Berechnung ist die Länge des Namens. Für jedes Zeichen des Strings wird nun folgende Berechnung durchgeführt:

– Wenn das Zeichen zwischen "a" und "z" liegt, wird es zunächst in Großbuchstaben gewandelt (-32).

– Der alte Hash-Wert wird mit 13 multipliziert, danach wird das Zeichen hinzuaddiert.

– Der daraus resultierende Wert wird auf die unteren 11 Bits reduziert (Und-Verknüpfung mit »$7FF«).

Der Hash-Wert des File-Namens ergibt sich nun aus dem Rest der Division des Wertes durch 72 (Anzahl von Hash-Zeigern in einem Directory-Block).

Handelt es sich bei der Diskette um eine Diskette im Format "DOS\2" oder "DOS\3", so wird nicht nur das normale Alphabet in Großbuchstaben gewandelt, sondern auch die Sonderzeichen wie zum Beispiel ä, ö oder ü.

(Red.)

Wo sind die Includes?

Um die Funktionen des OS 2.0 beim Programmieren nutzen zu können, benötige ich die zugehörigen Includes und Autodocs. Mein Händler, der über genannte Produkte nicht verfügt, hatte sich freundlicherweise bereiterklärt, bei Commodore nachzufragen. Das Ergebnis: In Deutschland ist kein Entwicklerpaket erhältlich!

Sollten Sie eine Möglichkeit ausfindig machen können, das Entwicklerpaket hierzulande zu erwerben, so wäre ich Ihnen dafür sehr dankbar.

Stefan Pröls, Schirmitz

Gute Nachricht: Ein Gespräch mit Commodore Deutschland ergab, daß das Paket nun auch direkt von Commodore erhältlich ist. Der Preis beträgt 45,– DM plus Porto und Verpackung, Lieferung per Nachnahme.

Die Adresse:
Commodore Büromaschinen GmbH
Lyoner Straße 38
D-6000 Frankfurt/Main 71
Telefon: 069-6638-0
Fax: 069-6638-229

(Red.)

Wo gibt's die "Omega Library"?

In der Ausgabe 1/92 beschreiben Sie die "Omega Library" der Firma Leo Capricorn aus Frankreich. Ich bin sehr an einer Bezugsadresse für dieses Produkt interessiert.

Karsten Majok, Albrechts

Die "Omega Library" ist erhältlich bei:
Pictures of Oz – Medienproduktion
Hochbrücker Mühle 2-3
5100 Aachen
Tel.: 0241-151109

(Red.)

Fragen zu den DTP-Programmen

In der Ausgabe 1/92 las ich in dem Artikel "Kaufberatung DTP" auf den Seiten 21 und 24 von mehreren DTP-Programmen für Amiga. Jetzt weiß ich aber leider nicht, von welchem Softwarehaus beziehungsweise Händler die folgenden Programme vertrieben werden. Könnten Sie mir bitte Adressen und Preise mitteilen?

Germo Schütz, Stuttgart

Da bei den genannten Programmen aufgrund des Dollar-Kurses sowie durch wechselnde Nachfrage die Preise nicht bei allen Anbietern gleich sind, bietet es sich an, anhand der Preislisten der verschiedenen Mail-Order-Firmen die günstigste herauszusuchen und dort zu bestellen. (Da es sich bei den aufgeführten DTP-Programmen durchweg um gängige Programme handelt, sollte das kein Problem sein.) Auf diese Weise kann man leicht einige Scheine sparen.

(Red.)

Amiga spielt nicht mit

Ich bekam das Spiel "Shinobi" von Sega geschenkt und wollte gleich damit spielen. Aber während des Spiels verschwand das Bild auf dem Monitor, der Ton war da. Könnte es vielleicht am Monitor liegen? Das Spiel kann es nicht sein, denn bei meinem Freund lief es perfekt. Bei anderen Spielen passierte mir das auch, aber nicht so oft. Kann es sein, daß mein Computer einer dieser Ausnahme-Computer ist?

Ein zweites Problem macht mir zu schaffen: Als ich nachschauen wollte, ob mein Computer Big- oder Fat-Agnus hat, sah ich bei den 512 KByte Chip-Memory eine 1-MByte-Option.

Geht es, daß ich vier von den neuen Megabit-Chips einlöte und noch vier Widerstände dazu, so daß ich dann 1 MByte Chip-Memory und 512 KByte Fast-Memory besitze?

Henry Christnick, Pahrenz

Das Problem mit dem fehlenden Bild dürfte auf Probleme mit dem Monitor deuten. Bei den Monitoren 1081 und 1084(S) treten häufig Fehler durch kalte Lötstellen beim Zeilentrafo auf, die zu den beschriebenen Bildausfällen führen können. Ein kleiner Test kann da unter Umständen Klarheit bringen: Durch leichtes Klopfen auf die linke Seite des Monitors kann das Bild unter Umständen wieder zum Erscheinen gebracht werden. Auf jeden Fall empfiehlt es sich, den Monitor zum Händler zu bringen, um ihn dort reparieren zu lassen, eventuell schon mit dem Hinweis auf den Zeilentrafo.

Weiterhin ist es nicht möglich, den A500 durch Einlöten von Speicherbausteinen auf 1,5 MByte zu erweitern. Die A500, die über die Möglichkeit des Einbaus weiterer Speicherchips verfügen, können außerdem nach dem Einbau nicht mehr durch Einstecken der Speichererweiterung A501 (oder baugleiche) um weitere 512 KByte erweitert werden. Da ein solcher Einbau außerdem mit mehr oder weniger großem Aufwand und Gefahren für den Rechner verbunden sein kann, sollte man lieber auf eine der zahlreichen 512-KByte-Erweiterungen zurückgreifen und die Löt-Lösung vergessen.

(Red.)

Meinung zur AMIGA DOS

Ich möchte Ihnen meine Meinung zur AMIGA DOS mitteilen:

– Abgedruckte Listings beziehen sich in der Regel auf kommerzielle Compiler (sind also in bezug auf das Taschengeld eines Zwölf- bis 18jährigen zu teuer). Um dieser Gruppe nicht den Spaß am Programmieren zu nehmen, sollten Hinweise zum Übersetzen mit PD-Compilern vorhanden sein: Wo gibt es den Compiler? Woher bekommt man die Includes und eventuell notwendige Libs? Und, sofern möglich, welche Switches sind zu setzen?

– Das Prinzip Databox ist gut. Aber die Disketten sind einfach zu teuer! In einem Jahr summiert sich das auf rund 300 DM. Wäre es nicht sinnvoller, einen Teil der Programme beziehungsweise Spiele auf die FD-Disketten zu tun und für den Rest den Preis zu halbieren?

Uwe Teegen, Braunschweig

Bei dem Abdruck von Programmen in der AMIGA DOS werden zunächst einmal die Angaben der Programmautoren als Basis genommen. Diese kennen sich – wie auch die Redakteure – meist nur mit "ihrem" Compiler aus, können also in den wenigsten Fällen Angaben machen, wie ihr Aztec-Programm unter DICE oder SAS C zu compilieren ist. In manchen Fällen ist eine Umsetzung auf einen PD-Compiler auch nicht möglich, weil Compiler-spezifische Funktionen verwendet wurden. Daher bleibt für Anwender von PD-Compilern zur Zeit leider nur, das Programm unter ihrem Compiler compilieren zu lassen und auf die Fehler zu warten. Anhand der Anleitung zu dem jeweiligen Compiler sollte es möglich sein, die Fehlerquellen zu finden und passend zu ändern.

Zur Zeit gibt es eigentlich nur einen ernstzunehmenden PD-C-Compiler, nämlich DICE von Matt Dillon. Dieser enthält neben Compiler, Assembler und Linker sowie einigen anderen Tools und Dateien auch die kompletten Includes, sofern man sich registrieren läßt. Außerdem sind natürlich auch alle nötigen Linker-Libraries enthalten, die man zur Programmierung benötigt. Eine frei kopierbare Version von DICE ist auf gängigen PD-Disketten zu finden. Mit dieser Version sind schon viele der Möglichkeiten von DICE auszunutzen. Eine Voll-Version ist entweder direkt von Matt Dillon oder über den deutschen Verteiler Stefan Becker erhältlich:

Stefan Becker
Holsteinstrasse 9
5100 Aachen

Der Preis beträgt 85,– DM, man erhält außerdem ein weiteres Update.

Zu der Databox: Der Preis der Databox setzt sich nicht nur aus den Kosten für die beiden Disketten und die Verpackung zusammen. Es kommen noch Kosten für Erstellung und Kopieren dazu. Weiterhin erhalten die Autoren ja ein Honorar für ihre Programme, das teilweise auch mit den Einnahmen der Databox bezahlt wird. Somit entspricht ein Teil des Preises der Benutzungsgebühr für das Programm. Und nicht zuletzt spart man durch die Diskette die Zeit beim Eintippen, die ja auch von einigem Wert sein sollte.

(Red.)

Probleme mit dem Speicher

Ich habe Probleme mit einer Speichererweiterung A2058 von Commodore, bestückt mit 4 MByte Speicher, in einem A2000-B. Dabei kommt es in den oberen zwei MByte zu Datenverlusten, das heißt, Daten, die zum Beispiel auf der RAM-Disk in dem besagten Adreßbereich sind, werden unleserlich. Eine RAD, die sich in diesem Speicherbereich befindet, meldet Read-Write-Fehler. Dieses Problem tritt nicht auf, wenn sich die RAD im Bereich \$200000 bis \$3fffff befindet.

Sowohl ein von Commodore beigelegtes Testprogramm als auch ein selbstgeschriebenes, welche den Speicher mit Bitmustern beschreiben und wieder auslesen, zeigen keine Fehler in den RAM-Chips an. Auch ein Auswechseln der für diesen Adreßraum zuständigen Bustreiber brachte keine Änderung.

Manfred Klier, Eschwege

Zunächst sollte man sicherstellen, daß man die Speichertest-Programme im selben Betriebszustand (= Temperatur des Rechners) durchführt, in dem

auch die Fehler auftreten. Weiterhin besteht auch die Möglichkeit, daß ein Virus in dem jeweiligen Bereich am Werk ist. Technisch gesehen gibt es eigentlich keinen Grund dafür, daß die Daten zwar verfälscht werden, aber beim Testen die Fehler nicht auftreten. Eventuell sollte das betreffende Testprogramm einige Zeit warten, bis es die geschriebenen Daten liest.
Unter Umständen kann es helfen, wenn man die Karte in einen anderen Slot steckt, da das Timing bei einigen A2000-Versionen nicht immer sehr verläßlich ist und dadurch mit einigen Karten Probleme auftreten.

(Red.)

Animations-problem

Folgende Frage ergibt sich im Zusammenhang mit den Animationsmöglichkeiten von »Deluxe Paint IV«: Wie kann man eine DPaint-IV-Animation startfähig auf eine Diskette kopieren?

Werner Klinkhammer, Niederaußem

Dazu muß man eines der im PD-Bereich üblichen Abspielprogramme wie »PlayAnim« oder »ShowAnim«, von denen sich im übrigen auch eines auf der Original-DPaint-Diskette befinden sollte, auf die Diskette kopieren und den Aufruf des Programms in der »Startup-Sequence« eintragen.

(Red.)

A500 Plus will nicht spielen

Ich bin enttäuscht! In Ihrer Zeitschrift ist immer wieder zu lesen, wie toll dieser Amiga ist und was er alles kann. Das ich nicht lache! Ich habe mir kurz vor Weihnachten einen A500 Plus mit Monitor gekauft. Eigentlich wollte ich einen normalen A500, wie mein Arbeitskollege einen besitzt, doch der Verkäufer des Kaufhauses, bei dem ich den Computer gekauft habe, riet mir zu diesem neuen Gerät, weil der alte A500 nicht mehr gebaut werden würde und der neue viel besser sei. Er hätte mehr Speicher und eine bessere Grafik und außerdem würden alle Programme des "alten Fünfhunderter" auch laufen.
Ich habe mir dann gleich ein paar Spiele einpacken lassen, die mir mein Kollege empfohlen hatte, und bin mit dem Paket nach Hause gegangen. Und dann mußte ich feststellen, daß von den fünf Spielen, die ich mir für teures Geld geleistet habe, ein einziges lief. Ich dachte zuerst an defekte Disketten, probierte aber, stutzig geworden, die Spiele alle bei meinem Kollegen aus. Was soll ich sagen, alle Programme funktionierten hundertprozentig. Mein Kollege meinte nun, daß der A500 Plus nicht das gleiche Betriebssystem hätte wie sein A500 und daß die Fehler daher kämen. Wie dem auch sein, Eure so hochgelobte, angebliche Supermaschine Amiga hat mich mehr Geld gekostet, als notwendig war. Hätte ich mir lieber einen PC gekauft, wie mir eine Bekannte riet.
Es bedankt sich ein genervter und enttäuschter Leser.

Timo Rückter, Hanau

Bitte schön! Nur für was, das ist uns nicht so ganz klar. Folgendes ist passiert: Sie haben sich einen A500 Plus gekauft, dazu Spiele-Software, die nicht läuft. Dazu kann man doch nur sagen, daß die Schuld bei der Software zu suchen ist, nicht beim A500 Plus und nicht bei uns. Schauen wir uns doch das Problem genauer an:
Der A500 Plus besitzt ein neues Betriebssystem, genannt "OS 2.0". Auf den Standard-Amigas war bis jetzt die Version 1.3 dominierend. Betriebssystem heißt aber auch, daß man als Programmierer auf die Interna des Rechners zugreifen kann. Spiele werden meistens in Assembler geschrieben, der Schnelligkeit wegen. Diese Spiele werden meistens auf der Ebene der Prozessoren programmiert. Das wäre eigentlich kein Problem, wenn man davon ausgehen könnte, daß jeder Programmierer sich an die von Commodore vorgegebenen Richtlinien hält und darauf achtet, nicht zu viel ohne Betriebssystem zu machen. Die meisten Spiele halten sich aber nicht an diese Regel. So ist gerade jetzt zu merken, daß viele Spiele-Software beim A500 Plus streikt.
In Ihrem Falle würden wir die Software schnappen und zum Verkäufer gehen, um alles umzutauschen. Das manche Spiele-Software läuft, zeigt eigentlich, daß das, was wir hier beschrieben haben, wahr ist. Es gibt zum Glück noch Software-Firmen, die auf Richtlinien achten. Und "MS-DOSen" sind da auch nicht besser. Fast alle Spiele benötigen VGA-Grafikkarten. Daß dabei aber noch Karten bestimmter Hersteller nicht angenommen werden, zeugt doch auch nicht gerade von Programmierkunst, oder?

(Red.)

Die Problemecke

Ab und zu tauchen in den Leserbriefen Probleme auf, die wir beim besten Willen nicht lösen können. Hier sind Sie gefragt, liebe Leser und Leserinnen der AMIGA DOS.
Wenn Sie eine Antwort auf die in der Problemecke stehenden Fragen wissen, schreiben Sie uns unter dem Stichwort "Problemecke" – wir veröffentlichen Ihre Antwort dann in dieser Rubrik oder leiten sie, falsch gewünscht, an den Schreiber der "Problemzeilen" weiter.

Pascal-Probleme

Ich besitze jetzt schon etwa einen Monat Kickpascal 2.0, mit dem ich bis auf das Handbuch sehr zufrieden bin. Manche Dinge werden dort einfach nicht ausreichend erklärt, wie zum Beispiel die Systemprogrammierung. Aus diesem Grund fände ich es gut, wenn Sie in der AMIGA DOS eine (Kick-)Pascal-Ecke einrichten könnten, in der kleinere Programme stehen, Tips & Tricks und Fragen und Antworten zum Thema Pascal.
Jetzt zu den Fragen:
1.) Wie reserviere ich Speicher für den Befehl »Flood« (»graphics.library«)?
2.) Wie kann ich unter Benutzung der »graphics.library« Grafiken auf dem CLI oder der Workbench ausgeben (in diesem Fall fiel der Begriff "RastPort-Bestimmung")?
3.) Wie kann ich in einem selbstdefinierten Window eine andere Schrift laden und setzen?

Martin Bücker, Hamm

Zum Thema "Kickpascal" noch ein Hinweis: Wenn wir Tips & Tricks zu Kickpascal bekommen und diese dem Programmierer tatsächlich helfen, dann werden sie auch in der AMIGA DOS abgedruckt. Leider tut sich da im Augenblick sehr wenig. Also, Kickpascal-Leute, meldet Euch! *(Red.)*

Die Maus ist (fast) tot

Ich habe ein Problem mit meiner Amiga-Maus, das Problem ist recht einfach beschrieben: Das elektronische Nagetier will nicht mehr. Dies äußert sich dadurch, daß zwar Bewegungen noch möglich sind, der Mauszeiger auch folgt, jedoch immer öfter stockt und stehenbleibt. Was kann man dagegen tun?

Hermann Roth, Bad Brückenau

Das sieht ganz danach aus, als ob das Nagetier nicht im Sterben liegt, sondern eher mal "gebadet" werden müßte. Diese Fehler tauchen dann auf, wenn sich an den Rollen Fuseln und Staub zu einer mechanischen Bremse verbinden. Da die Rollen die Lichtschranken-Unterbrecher bewegen, kann ein Stocken die beschriebenen Auswirkungen haben. Deshalb sollten Sie Ihren "Mäuserich" einmal aufschrauben und die Rollen mit einer Pinzette an den Enden "entflusen", das heißt an den Rollenrändern säubern.

(Red.)

Vesalia Computer

Industriestraße 25 • 4236 Hamminkeln
Tel. 02852/1068/69/60 • Fax 02852/1802
Mo. - Fr. 9 - 17 Uhr
Autobahn A 3, Oberhausen - Arnheim:
Ausfahrt Wesel/Bocholt

Filecard und Harddisk für Amiga 500 - 3000

Datendurchsatz von über 2 MB/Sekunde möglich. Alle Filecard und Harddisk komplett anschlußfertig. Bereits formatiert und mit WB 1.3 installiert Harddisk für Amiga 500 und A 1000 als Monitoruntersatz. Winner II SCSI Test Kickstart 7/8 1991 „Sehr Gut"

Winner II SCSI 16 bit Filecard A 2000/3000
52 MB LPS 1,11 MB/Sek. **889,-**
105 MB LPS 1,11 MB/Sek **1148,-**
Winner II Controller **348,-**

Multi-Evolution A 500 Controller mit RAM-Option
52 MB LPS **889,-**
105 MB LPS **1148,-**
Controller **348,-**

Winner I für Amiga 500 und Amiga 1000. Autoboot ab Kick 1.2.
31 MB RLL 698,- 42 MB RLL 798,- 63 MB RLL 898,-

512 KB-Ram A 500 intern **69,-**
abschaltbar mit Uhr und Akku
2 Jahre Garantie.

1 MB-Ram A 500 Plus intern **149,-**
2 Jahre Garantie.

2 MB-Ram A 500 intern **99,-**
Mit 512 KB bestückt, abschaltbar.
Mit Uhr/Akku. 2 Jahre Garantie.

2 MB-Ram A 500 intern **285,-**
abschaltbar. Mit Uhr/Akku.
2 Jahre Garantie.
CPU-Adapter **48,-**

8/2 MB-Ram A 2000 intern **345,-**
4,0 MB -Ram " 2000 intern **545,-**
8,0 MB -Ram " 2000 intern **965,-**

3,5" Winner-Drive **145,-**
Mit durchgeführtem Bus bis DF 3,
abschaltbar. Metallgehäuse.
Made in Germany. 1 Jahr Garantie

3,5" Color-Drive **159,-**
Mit durchgeführtem Bus bis DF 3,
abschaltbar, Metallgehäuse.
Made in Germany. 1 Jahr Garantie

5,25" Winner-Drive **199,-**
Mit durchgefühtem Bus bis DF 3,
abschaltbar und mit WP-Schalter

5,25" Amiga 2000 intern **229,-**
Komplett mit Interface und Bootselector. 1 Jahr Garantie.

3,5" DF 2 Amiga 2000 intern **199,-**
inkl. 5,25" Einbaurahmen und
Interface. 1 Jahr Garantie.

3,5" Amiga 2000 intern **129,-**
Komplett mit Einbaumaterial. 1 Jahr Garantie.

3,5" Amiga 500 intern **139,-**
Komplett mit Auswurftaste.
1 Jahr Garantie.

Elektronischer Bootselektor **48,-**
wahlweise booten von DFO - DF 3,
das interne LW ist abschaltbar.

Sounddisketten Nr. 1 - 7 je **9,-**
alle Disketten **49,-**
Nr. 2 - 6 Instrumente Nr. 1
Drum-Section Nr. 7 Bass & Sounds

Winner-Mono-Soundsampler **79,-**
Samplingrate 55930 Hz.
Mit Software.

Winner-Stereo-Soundsampler **129,-**
Samplingrate 23243 Hz. Mit Software.

Winner-Midi A 500 - 3000 **79,-**
In. Thru und 2 x Out, schneller
Optokoppler, schönes Metallgehäuse.

Winner-Midi-Plus A 500 - 3000 **139,-**
In. Thru und 3 x Out, Metallgehäuse
Optokoppler mit 700 % Kopplungsfaktor, Leistungstreiber an allen Ausgängen.

Disketten 100 % errorfrei
3,5" 2 DD NN 10 Pakete **90,-**
5,25" Disketten NN 100 Stück **50,-**

Pal-Genlock 2.0 **648,-**
von Elektronic-Design.

Y-C Genlock **998,-**
Jetzt noch bessere Bandbreite und
zusätzlichem Anschluß für S-VHS
und Hi 8. RGB-Bandbreite 10 MHz.

Y-C Splitter **498,-**
Bandbreite besser als PAL-Standard.

Dig-Splitt-Junior **278,-**
Elektronischer Farbsplitter, auch
für Hi 8 und S-VHS. Mit Anschlußkabel.

Video-Split-IT **298,-**
RGB-Splitter mit Monitorausgang,
LED-Anzeige und allen Anschlußkabeln.

Interlace-Card A 2000 **288,-**

Interlace-Card A 500 **298,-**
Kein Flimmern, keine schwarzen
Streifen. Mit Stereo-Ausgang, Software und Einbauanleitung. 1 Jahr
Garantie. Direktanschluß für VGA- und
Multisync-Monitore.

VGA-Farb-Monitor 14" **748,-**
30 MHz Bandbreite

ROM-ROM Umschaltplatine **59,-**
Für Kick 1.3 und 2.04

Kickstartrom 1.3 **59,-**
Kickstartrom 2.04 **99,-**
Denise **63,-**
ECS Denise **78,-**
Garry **49,-**

8520 **49,-**
BigFat-Agnus **99,-**
A 500 Netzteil 4,5 A **99,-**
Das besonders Starke.

Autoboot-Modul A 2000 **99,-**

A2090-Autoboot-Modul **159,-**

A2090 A-Turbo-Chip-Satz **149,-**

Seagate-MFM oder RLL-Controller **99,-**
Kabelsatz **8,-**

3,5" HD-Träger **7,50**
OMTI-Seagate-Adapter A 2000 **69,-**

Autoboot-Set A 2000 **298,-**
komplett mit Seagate-Controller,
Kabelsatz und Winner-Soft (autoboot)

Autoboot-Set A 500/1000 **298,-**
komplett mit Seagate-Controller,
Adapter mit Busdurchführung,
Winner-Soft, autobott 1.2.

Festplattengehäuse **59,-**
Schaltnetzteil **109,-**

MouStick Amiga 500/3000 **44,-**
MouStick Amiga 2000/2500 **49,-**
Automatische Umschaltung zwischen
Mouse-Mouse, Joystick-Joystick oder
Mouse-Joystick. 2 Jahre Garantie.

Amiga-Bremse intern **39,-**
Amiga-Bremse ext. A 500 **65,-**

PowerFire **19,-**
Dauerfeuer-Interface

Philips Stereo Monitor **548,-**
kompl. mit Anschlußkabel

DIN A 3 Drucker **259,-**
inkl. Druckerkabel

Winner-Mouse **49,-**
In weiß, rot, gelb oder schwarz

Winner-Mouse **59,-**
inkl. Mouse-Halter und Pad.
In weiß, rot, gelb oder schwarz

Swifty-Mouse **49,-**
Amiga/Atari schaltbar
Trackball **89,-**
Multiscan-Farb-Monitor 14" **1048,-**
Auch für Amiga 3000, inkl. Kabel.

Test

Platinenlayout-Programm

Pin-Board

Hobby-Elektroniker kennen das Problem, mal eben eine Platine zu entwickeln und zu ätzen. Wenn nur das blöde Kleben nicht wäre. Maxons »PLP« verspricht da Abhilfe.

Der Markt der Platinenlayout-Programme für den Amiga ist ja nicht gerade überfüllt zu nennen. Neben dem alten Klassiker »NewIO« und »ProBoard«, die beide für den Hausgebrauch ein wenig teuer sind, gibt es nur das wenig taugliche »PCLO-CAD«, welches von den Möglichkeiten und der Qualität her nicht gerade berauschend ist (und auch wohl kaum noch zu bekommen sein dürfte). Die Firma Maxon hat kürzlich ihr Programm »PLP« herausgebracht, daß diese Lücke schließen soll.

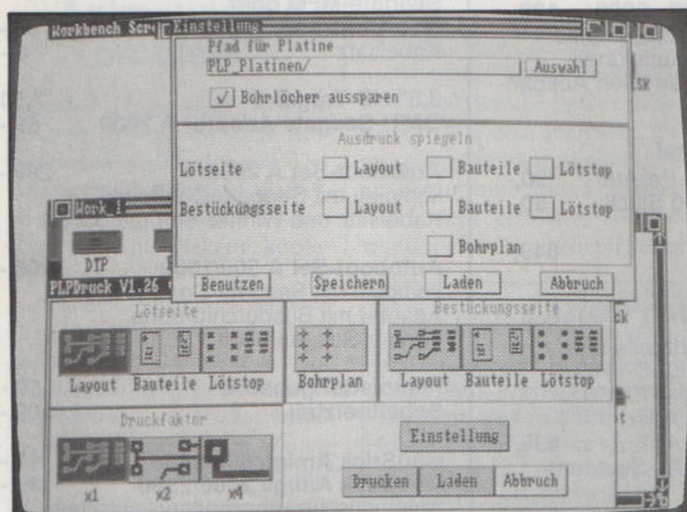


Bild1. »PLP-Druck« übernimmt den Ausdruck des Layouts

Technische Daten

Im Lieferumfang befindet sich neben einem knapp 60seitigen Handbuch eine Programmdiskette, die neben dem Layoutprogramm und Drucker- und Plottertreibern verschiedene Beispielplatinen und einige Bauteile enthält. Das Programm kann komplett auf die Festplatte kopiert werden. Außerdem ist es sowohl unter AmigaOS 2.0 als auch auf A3000 voll lauffähig. (Während des Tests auf A3000/2.0 und A1000/1.3 waren keinerlei Abstürze zu beklagen.)

In der vorliegenden Version ist die Größe der zu bearbeitenden Platine nur durch den freien Chip-Speicher begrenzt. Auf einer 2-MByte-Chip-Maschine sind dabei Platinen mit etwa 52x52 cm möglich. Bei 512 KByte reduziert sich dies auf etwa 20x20 cm. Dies dürfte für die meisten Platinen voll ausreichend sein (Eurokarten haben bekanntlich eine Größe von 160x100 mm). Neben der Platinengröße ist auch die Anzahl der Bauteile nur durch den freien Speicher begrenzt. »PLP« kann man – wenn man denn will – komplett über die Tastatur steuern. Die Tastaturbelegung ist dabei anfangs zwar wenig einprägsam, aber schon nach der ersten Platine sollte man so firm sein, daß alle Arbeiten flüssig erledigt werden können. Unangenehm fällt nur auf, daß die deutsche Tastaturbelegung fest einprogrammiert ist, weshalb sich Benutzer einer US-Tastatur nicht ganz auf die Zeichen, sondern teilweise auf die Position der Tasten verlassen müssen – Besitzer der älteren A1000 müssen da leiden.

Erste Layouttests

Im Rahmen der ersten Gehversuche mit »PLP« wurden zunächst einmal die mehr oder weniger sinnvollen Beispiele ausprobiert, die auf der Diskette enthalten sind. Die erste positive Überraschung: »PLP« ist in der Lage, ein Bussystem von mehreren ICs problemlos auf einer Seite der Platine zu routen (wobei andere Layoutprogramme mitunter Probleme haben). Außerdem läßt die größere Platine für die eigene Zukunft hoffen.

Nun folgt ein weiterer Test, und zwar diesmal des Handbuchs, denn ohne dieses wird man wohl eine Weile brauchen, bis man dieses leistungsstarke Programm beherrscht. Auf den ersten Blick scheint es mit seinen 57 Seiten etwas dünn geraten, was sich auch auf den zweiten Blick nicht ändert. Es werden zwar fast alle Funktionen gut und ausführlich erklärt, doch fehlt der Anleitung ein wenig der rote Faden. Ein Kapitel mit einer kurzen Beispielsitzung könnte dem neuen Benutzer auch gut tun. So muß man sich beim Erstellen seiner ersten Platine doch ständig durch das Inhaltsverzeichnis und die verschiedenen Kapitel wühlen, um die passenden Tasten zu finden.

Ein Testausdruck mit dem Druckprogramm ist – wenn man »NewIO« kennt – etwas überraschend. Statt wie »NewIO« nur einige Drucker fest eingebaut zu haben, verwendet »PLP« die Druckertreiber der Workbench, womit es für nahezu jeden Drucker einsetzbar ist. Nur sollte man nicht vergessen, die Ausgabequalität möglichst hoch einzustellen.

Am meisten vermag ein Ausdruck auf einem Laserdrucker zu überzeugen. Wird dieser dann noch in der doppelten Größe erstellt, erhält man nach einer Verkleinerung und Kopieren auf Folie eine Vorlage, die sich zum Belichten hervorragend eignet. Auch stimmen die Ausmaße und Abstände der Pins exakt (auch eines der Mankos unter anderem bei »NewIO«; hier konnte es passieren, daß die letzten zwei oder drei Pins eines großen Sockels nicht mehr so ganz richtig paßten). Ein Treiber für einen Plotter ist ebenfalls enthalten, bei diesem müssen die Befehle zum Plotten einzeln eingegeben werden. Nadeldrucker, vor allem 24-Nadler, bringen aber auch sehr gute Ergebnisse.

Noch Bauteile gefällig?

»PLP« arbeitet auf der Basis von verschiedenen Bauteilbibliotheken, die vom Benutzer je-

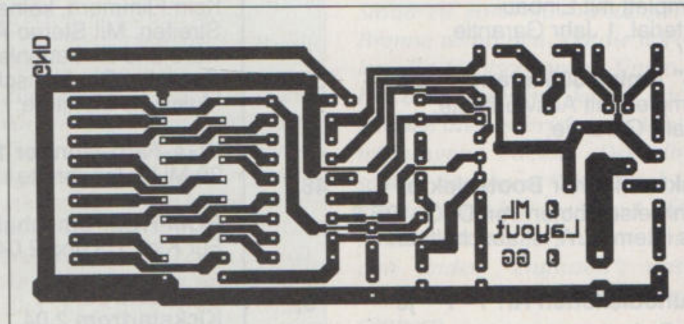


Abb. 1. Ausgedruckte fertige Vorlage (HP Laserjet)

derzeit erstell- oder änderbar sind. Durch das Ansprechen über die Zahlentasten sind die zehn Teile pro Bibliothek leicht erreichbar.
Neben den Bibliotheken, die man auf die Funktionstasten legt (frei definierbar), sind jederzeit weitere Bibliotheken einladbar. Somit kann man beliebig viele verschiedene Teile auf einer Platine verwenden.
Der Bauteileditor funktioniert genau wie der normale Platineneditor auch. Man kann Lötpunkte, Leiterbahnen und Texte setzen und Teile anderer Bauteile einfügen (allerdings nur innerhalb einer Bibliothek). Durch die Rasterabstände von 0.3 mm sind hier auch SMD-Bauteile erstellbar (Surface Mounted Devices = oberflächenmontierbare Bauteile; sind äußerst kompakt und klein und werden häufig dort eingesetzt, wo viel Elektronik auf kleinstem Raum benötigt wird). Positiv fiel dabei noch auf, daß »PLP« automatisch die Ausmaße des Bauteils erkennt.

Platinenlayout leichtgemacht

Jedem Bauteil kann man im Editor eine zweizeilige Beschreibung zuordnen; außerdem hat man die Möglichkeit, auch jedes einzelne Pin mit einer Kennung zu versehen, was die spätere Verwendung erleichtert.

Im letzten Test mußte »PLP« nur noch beweisen, wie praxistauglich es ist. Dazu wurde eine kleine Platine mit zwei ICs und verschiedenen Kleinteilen erstellt (vergleiche Abb. 1). Dabei war nicht nur die normale Handhabung, sondern auch die Qualität des Routers gefragt.
Bei der Plazierung der Bauteile ist positiv anzumerken, daß sowohl Bauteile als auch die an sie angeschlossenen Verbindungen jederzeit neu zu positionieren sind. Auch das Drehen eines Bauteils ist mit keinen Problemen verbunden.
Bei dem Vergeben der Rout-Verbindungen ist Voraussetzung, daß jedes Bauteil eine Bezeichnung hat. Leider wird vom Programm eine Numerierung oder Bezeichnung nur durch eine bestimmte Tastenkombination ermöglicht; ein Umstand, der gleich zu Anfang Probleme ergeben kann, wenn man sich keine "Spickzettel" an die richtige Stelle hängt.
Hat man erst einmal einige Verbindungen mit dem Netzeditor erstellt, so kann man darangehen, diese zu routen. Wie bei jedem einfacheren Router, empfiehlt es sich auch hier, das Routen der Platine etappenweise zu machen, so daß man, falls nötig, kleinere Änderungen an den Ergebnissen vornehmen kann. Generell ist das Ergebnis des Routers recht gut; es fehlt leider die Möglichkeit, ihn auf die Löt- oder Bestückungsseite zu begrenzen, wodurch Verbindungen öfters mal statt über einen Umweg auf der einen Seite auf der anderen Seite gelegt werden.
Sind die Verbindungen gelegt, so kann man diese benennen. Dabei wird dann automatisch der gesamte Rout-Weg korrekt belegt. Wenn man diese Leitung mit der Maus berührt, wird der Name dieses Signals am Bildschirm angezeigt (wie auch bei den Bauteilen und den Bauteil-Pins). Dies sorgt dafür, daß man bei größeren Platinen den Überblick behält.

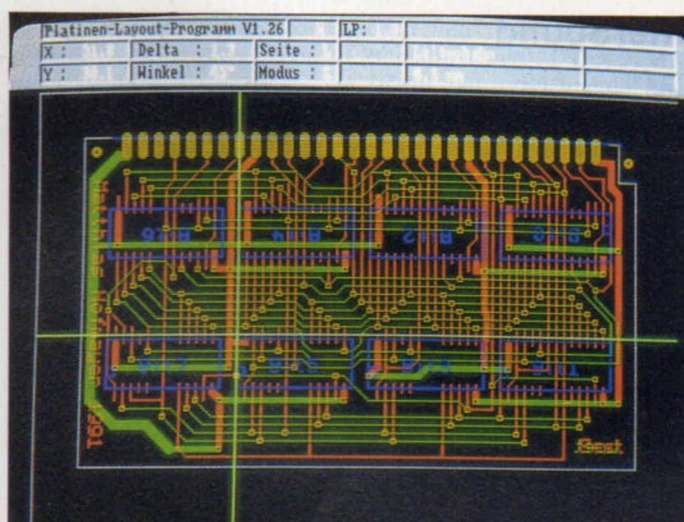


Bild 2. Die Arbeitsoberfläche zu »PLP« ...

Besser als Kleben

Für die Erstellung der Platine ist wichtig, daß die erzeugte Platine nachträglich noch auf fehlende Verbindungen oder Kurzschlüsse überprüft werden kann, was durch Nachbearbeitung per Hand leicht vorkommen kann. Dazu werden alle im Netzeditor eingegebenen Verbindungen auf den vorliegenden Wegen geroutet und mit den erlaubten Verbindungen verglichen.
Ist die Platine einmal vervollständigt, kann man zum Schluß noch den Bestückungsplan erstellen. Hierbei unterstützt »PLP« die Arbeit ebenfalls recht gut. So kann man mit einem Tastendruck die Bezeichnung eines Objekts übernehmen und plazieren. Weiterhin sind natürlich auch andere Texte möglich, die in verschiedenen Größen verwendet werden können.
»PLP« ist bereits in der ersten Version ein gutes Produkt, welches das Kleben von Platinen endlich der Vergangenheit angehören läßt. Der Preis von rund 250 DM ist bei der Leistung und Qualität eher untertrieben und macht sich schon nach kurzer Zeit bezahlt.

Garry Glendown (jb)

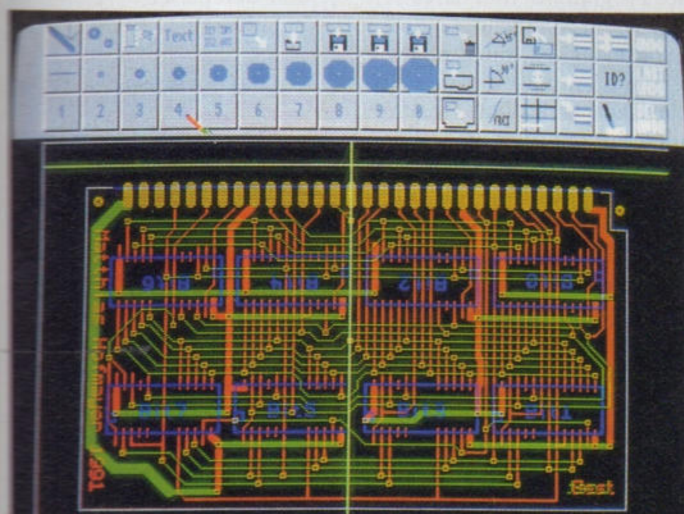


Bild 3. ... und der "Arbeitstisch"

PLP

AMIGA DOS

Testurteil 3'92

gut	**82%**
Anleitung:	60%
Installation:	70%
Bedienung:	80%
Tauglichkeit:	85%
Preis/Leistung:	90%

Hersteller: Maxon
Testmuster: Maxon
Preis: 248,– DM
Konfiguration: alle Amiga, Platinengröße richtet sich nach Speicher

InterSound

Samples made in Dänemark

Ein Programm zum Editieren von digitalisierten Instrumenten, das laut Handbuch "schnell und zuverlässig" ist und "neue, hochentwickelte Funktionen" besitzt? Das macht doch irgendwie neugierig.

Amiga-Besitzer sind verwöhnt, was Soundsampler und entsprechende Programme betrifft. »AudioMaster IV« und »Audition 4« haben Maßstäbe im Bereich der Audio-Digitizer gesetzt, die nur schwer zu übertreffen sind.

Neandertal-Programm?

Viele Programme erreichen diese Maßstäbe sogar nicht einmal annähernd. Leider ist »InterSound« von Frits Nielsen solch ein Programm. Vom Hersteller als "professionelles Samplerprogramm" für Mono- und Stereo-Sampler bezeichnet, macht es bereits bei ersten Sample-Versuchen nur zu deutlich, daß es in Wahrheit ein oder zwei Klassen tiefer einzuordnen ist: Die höchste Abtastfrequenz beim Herstellen von Samples beträgt etwas mehr als 14000 Hertz. Das ist für die meisten Anwendungszwecke, die über bloßes Herumspielen hinausgehen, zu wenig.

Etwas besser sieht es bei den Bearbeitungsmöglichkeiten aus. Die grundlegenden Funktionen wie Ausschneiden, Kopieren und Einfügen von Sample-Bereichen sind alle vorhanden. Sehr angenehm fällt dabei auf, daß »InterSound« bis zu acht Samples gleichzeitig im Speicher hält, zwischen denen der Benutzer mit einem simplen Mausklick hin- und herwechseln kann. Jedem dieser Samples sind wiederum acht "Segmente" (markierte Bereiche des Samples) zugeordnet, die sich ebenfalls mit der Maus anwählen lassen. Bis zu vier der Samples können gleichzeitig in eigenen Fenstern dargestellt werden (siehe Bild). Das vereinfacht es erheblich, Teile eines Sounds aus einem Sample in ein anderes zu kopieren.

Die Effekte, die auf markierte Sample-Bereiche angewandt werden können, sind gute Hausmannskost (Fading, Hall, Echo, Vibrato, Tremolo), gekrönt von einigen Schmankerln (Resamplen auf eine bestimmte Tonhöhe, Mischen zweier Samples, Stereokanäle schrittweise miteinander vertauschen). Alle diese Funktionen laufen in akzeptabler Geschwindigkeit ab. Raffiniertere Funktionen wie Mehrfachschleifen, digitale Bandfilter oder gar Oversampling sucht man allerdings vergeblich.

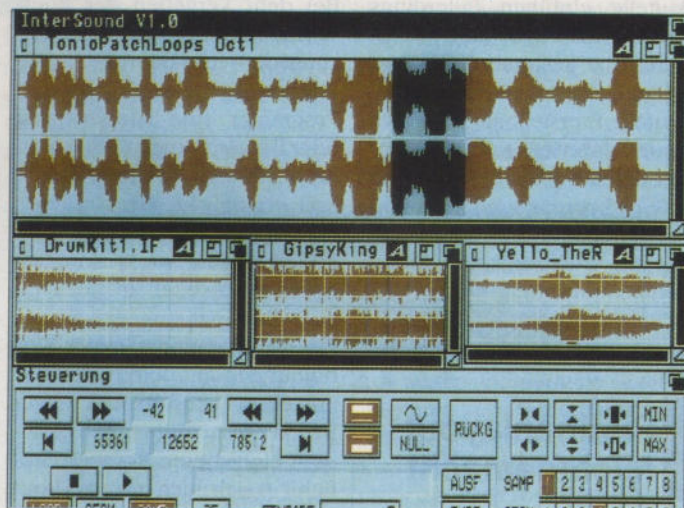


Bild 1. »InterSound« kann bis zu acht Samples gleichzeitig verwalten

Alle grundlegenden Funktionen sind also zufriedenstellend und besser.

Auch eine ganze Reihe von ansprechenden Effekten ist vorhanden. Der große Pluspunkt des Programms, und damit seine Daseinsberechtigung, ist seine Fähigkeit, nicht nur wie »Perfect Sound« mehrere Samples zu verwalten, sondern auch bis zu vier von ihnen gleichzeitig darzustellen. Der Bedienungskomfort von »InterSound« läßt allerdings sehr zu wünschen übrig.

"Altertümchen"

Einsteiger, die lediglich ein bißchen mit ihrem neuen Sampler herumspielen möchten, werden schon mit diesem Programm erste Erfolge erzielen können. Dem steht aber leider die Tatsache entgegen, daß das »InterSound«-Handbuch eines der verwirrendsten und fehlerhaftesten ist, die mir je untergekommen sind. Außerdem gibt es in der Public Domain eine etwas betagte Version des »Perfect Sound«, Version 2.1, die immerhin mehr zu bieten hat – und das für sehr viel weniger Geld.

»InterSound« ist also für Hobby-Zwecke durchaus brauchbar, ein "wirklich leistungsstarkes Werkzeug für die einmaligen Sampling-Möglichkeiten des Amiga-Systems", wie die Anleitung so optimistisch verkündet, ist das Programm jedoch eindeutig nicht. Wer mit seinem Sampler mehr machen will, sollte sich die Tests über »AudioMaster IV« oder »Perfect-Sound« durchlesen. ❑

Hartmut Schumacher (jb)



Bild 2. Leider nicht zeitgemäß: »InterSound«

Amiga-DOS-Info

Da »InterSound« noch keine deutschen Vertreiber hat, hier erst einmal eine Vorab-Wertung.

positiv:
- einfach zu bedienen
- bis zu acht Samples gleichzeitig

negativ:
- schlechter Bedienungskomfort

InterSpread

Verrechnet!

Denkt man an Amiga, denkt man kaum gleichzeitig an Tabellenkalkulation. Grund hierfür sind die wenigen Tabellenkalkulationen, die es für den Amiga überhaupt gibt.

Bei häufig wiederkehrenden Rechenoperationen läßt es sich nicht vermeiden, auch auf dem Computer immer wieder die gleichen Eingabeabläufe zu machen. Helfen können hier Tabellenkalkulationen. Tabellenkalkulationsprogramme stellen dem Benutzer ein elektronisches Arbeitsblatt zur Verfügung, in dessen Felder (Zellen) er Werte eintragen kann (siehe Bild 1). Mehrere Zellen lassen sich miteinander verknüpfen, so daß beispielsweise eine Zelle die Summe der Werte zweier anderer Zellen enthält. Ein solches elektronisches Arbeitsblatt hat gegenüber einem simplen Blatt Papier einen unschätzbaren Vorteil:
Wenn der Benutzer einen Wert in einer Zelle verändert, berechnet das Programm automatisch die Werte aller mit ihr verknüpften Zellen neu. Das Durchspielen mehrerer Varianten einer Situation unter verschiedenen Ausgangsvoraussetzungen wird dadurch zu einem Kinderspiel.
Für den Amiga existieren bereits zwei gute Tabellenkalkulationen, »MaxiPlan« und »The Advantage«, an denen sich ein neuer Herausforderer messen muß.

Zahlenakrobatik

»InterSpread« von Henrik Krogh Moeller kann bis zu zehn Tabellen gleichzeitig verwalten und zwischen ihnen Daten austauschen. Die Zellen lassen sich mit über 60 Funktionen miteinander verknüpfen, darunter nicht nur die Grundrechenarten, Finanz- und Datumsfunktionen, sondern auch trigonometrische Funktionen.
Um Aufgaben zu automatisieren, die viele einzelne Eingaben erfordern, lassen sich diese Eingaben in Makros zusammenfassen. Auf diese Weise können auch Benutzer, die keine Kalkulationsexperten sind, komplexere Arbeitsblätter benutzen.

Grafiken statt vieler Worte

»InterSpread« kann zwar ASCII-Dateien exportieren und Daten mit dem Textverarbeitungsprogramm »InterWord« des gleichen Herstellers (siehe Test in dieser Ausgabe) austauschen, Dateien anderer Tabellenkalkulationen, vor allem der PC-Welt (neben den oben genannten Programmen beispielsweise »Lotus 1-2-3«), lassen sich jedoch bedauerlicherweise nicht laden.

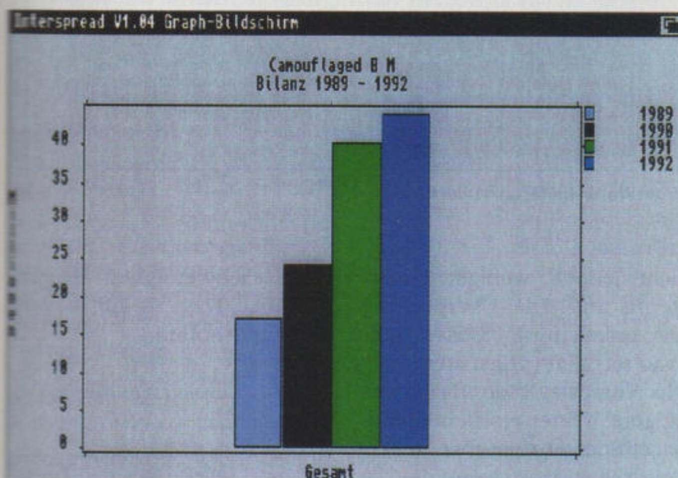


Bild 2. Interessant: die grafische Ausgabe

InterSpread V1.04 Project: CBM — Feld: F20 — M:Klar
Eingabe: =F8-F15 — I:Klar

		C	D	E	F
1	CBM - Camouflaged Business Machines - Bilanz				
3		1989	1990	1991	1992
5	Hardware	20	28	45	64
6	Software	13	17	29	25
8	Plus	33	45	74	89
11	Gehälter	11	14	22	28
12	Support	2	4	5	8
13	Spenden	3	3	7	9
15	Minus	16	21	34	45
20	Gesamt	17	24	40	44

Bild 1. »InterSpread«: stark verbesserungsbedürftig

Die in der Tabelle berechneten Daten können anschließend – zwecks besserer Übersicht – grafisch dargestellt werden. »InterSpread« bietet die üblichen Diagrammarten wie Linien, Säulen, Stufen und Torten, ist jedoch hinsichtlich verschiedener Variationen dieser Diagramme weniger flexibel als seine Konkurrenten.
Leider können die fertigen Grafiken nicht in einem Vektorformat (wo die Berechnung der Grafik nicht nach Pixeln, sondern nach berechneten Koordinaten stattfindet) abgespeichert werden, sondern nur als IFF-Dateien. Dies führt bei der Verarbeitung der Grafiken mit DTP-Programmen zu den häßlichen und unprofessionellen Bitmap-Zacken ("Treppchen-Effekt" bei gezeichneten Linien).
Alle diese positiven und negativen Eigenschaften des Programms sind in der vorliegenden Version 1.04 jedoch gegenstandslos. So unglaublich es klingt: Das Programm ist nicht in der Lage, seine eigenen abgespeicherten Dateien wieder einzulesen! Dahingehende Versuche enden mit der lakonischen Mitteilung: "Verzeihung – Die Tabelle ist von einer nichtkompatiblen InterSpread-Version". Ein weitgehendes Update könnte aus »InterSpread« ein durchaus anschaffenswertes Programm machen. Zur Zeit kann es jedoch niemandem empfohlen werden – und über den "professionellen Anspruch" breiten wir besser den Mantel des Schweigens. ❑

Hartmut Schumacher (jb)

Amiga-DOS-Info
Da »InterSpread« noch nicht in Deutschland vertrieben wir, hier eine vorläufige Bewertung.
positiv:
- Grafikdarstellung
negativ:
- fehlerhaft

InterWord

Flink und einfach

»InterWord« ist der Name einer dänischen Textverarbeitung. Doch keine Sorge: Wer mit dem Programm arbeiten will, kann dies in deutscher Sprache tun.

Textverarbeitungen auf dem Amiga haben es schwer. Während auf dem PC Textverarbeitungen aller Art fast eine Vormachtstellung haben, sind die Amiga-Programme eingeschränkt professionell zu nennen. Entweder es harpert mit der Geschwindigkeit, oder das Programm hat ein paar Funktionen zu wenig. »InterWord« von Peter Kaae Thomsen ist eines derjenigen Textverarbeitungsprogramme, die ihren Namen wörtlich nehmen und sich nüchtern auf die Verarbeitung von Text konzentrieren. Grafikeinbindung oder Nutzung von Bitmap-Schriften sind nicht vorgesehen.

Da also bei der Darstellung des Dokuments keine recheninten-siven Grafikfunktionen berücksichtigt werden müssen, kann das Programm mit einer erfreulich hohen Arbeitsgeschwindigkeit aufwarten – ein nicht zu unterschätzender Vorteil gegenüber grafikunterstützenden "Schleichern".

»InterWord« besitzt die üblichen Funktionen, die jeder Editor hat: Löschen, Ausschneiden, Kopieren, Einfügen, Sprungbefehle, ASCII-Import und -Export. Darüber hinaus weist das Programm eine Reihe von Funktionen auf, die für ein Textverarbeitungsprogramm als Standard angesehen werden können (Suchen und Ersetzen, Kopf- und Fußtexte, Randeinstellungen, Tabulatoren).

Nichts Revolutionäres

Durch einige Eigenschaften hebt sich »InterWord« von einigen seiner Konkurrenten ab: mehrere Textfenster, Serienbriefe, Indexerstellung, Paßwortschutz, automatische Sicherheitskopien, Rechtschreibüberprüfung, automatische Trennfunktion, veränderbare Druckertreiber, Datenaustausch mit der Tabellenkalkulation »InterSpread« vom gleichen Hersteller sowie die Möglichkeit, neben dem linken Textrand Anmerkungen zu setzen.

Aber auch die Liste der Funktionen, die man vergeblich sucht, ist recht lang. Neben der fehlenden Grafikunterstützung vermißt der anpruchsvollere Benutzer vor allem die Möglichkeit des Spaltensatzes, eine Absatz- und Fußnotenverwaltung, einen "Thesaurus" (so nennt man Korrekturprogramme, die automatisch den Text auf richtige Schreibweise überprüfen) sowie Makros oder zumindest Floskeln.

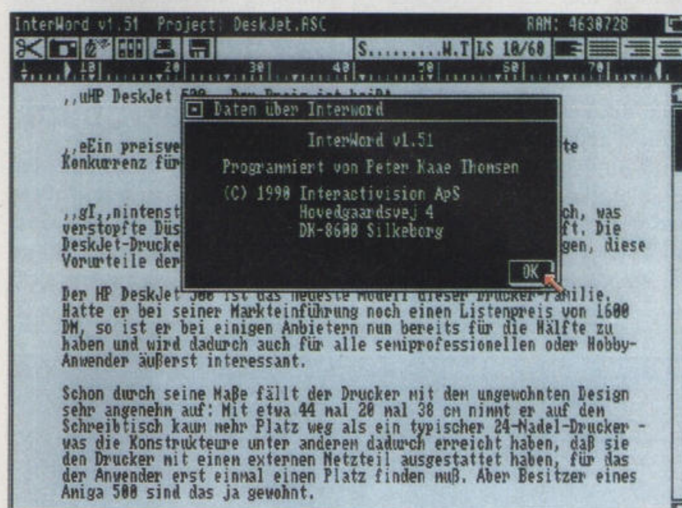
Nach Herstellerangaben umfaßt das Rechtschreiblexikon etwa 200000 Wörter. Das klingt recht beeindruckend, die Praxis sieht jedoch weniger begeisternd aus: Die Überprüfung auf unbekannte Wörter geht zwar recht flott vonstatten, aber die Vorschlaglisten für etwas längere Wörter erscheinen erst nach einer unzumutbar langen Zeit; selbst dann, wenn die Lexikondateien nicht von Diskette, sondern von Festplatte eingelesen werden.

Neuer Versuch aus Dänemark

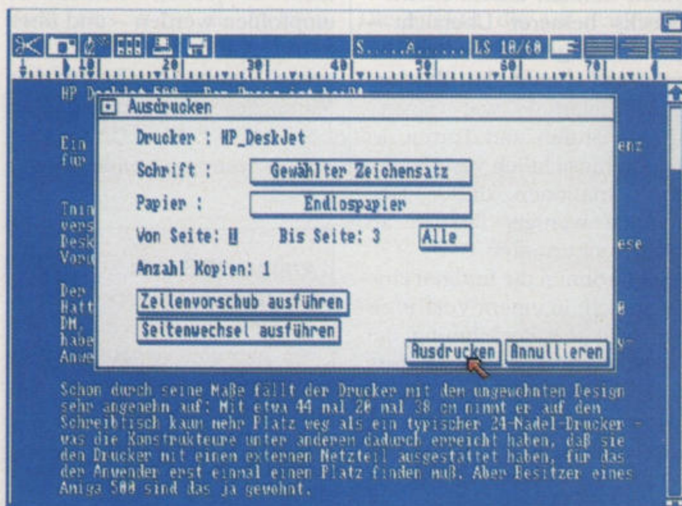
Die Worttrennung ist ebenfalls nicht die allerschnellste, dafür jedoch sehr zuverlässig. Sowohl die Rechtschreibüberprüfung als auch die Worttrennung können bereits automatisch während der Texteingabe ablaufen.

Wer auf eine hohe Arbeitsgeschwindigkeit Wert legt, dafür jedoch auf speziellere Funktionen wie eine Fußnotenverwaltung oder Grafikeinbindung verzichten kann, ist mit »InterWord« nicht schlecht beraten: Für Briefe oder auch für längere Texte ohne professionelle Gestaltungsansprüche ist das Programm ohne weiteres geeignet. Für Aufgaben im akademischen oder im DTP-Bereich hingegen kann es nicht empfohlen werden. Wie man schnell sieht, ist »InterWord« zwar auch nicht das »Ei des Kolumbus«, bietet aber genug Funktionen, und auch die Geschwindigkeit ist annehmbar. Das läßt für weitere Versionen hoffen. ❑

Hartmut Schumacher (jb)



»InterWord« bietet nicht viel Neues



»InterWord« hat eine nüchterne, aber praktische Benutzeroberfläche

AMIGA-DOS-Info

Da »InterWord« bei Redaktionsschluß noch keinen deutschen Vertreiber gefunden hatte, hier nur eine vorläufige Wertung.

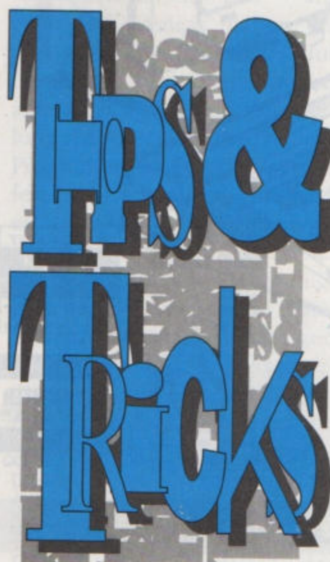
positiv:
- gute Installation
- einfache Bedienung
- gut im Gebrauch

negativ:
- Anleitung
- keine Grafikeinbindung
- kein Spaltensatz

Hardware

1084 sucht Anschluß

Jeder, der seinen Computer über den sogenannten Scart-Anschluß (oder auch Euro-AV-Buchse genannt) an den Monitor 1084 von Commodore angeschlossen hat, wird sich schon des öfteren über diese unsolide Verbindung geärgert haben.



Ich habe neben dem Amiga (an der Scart-Buchse) auch noch einen Videorecorder (als Fernsehtuner) an den Cinch-Buchsen, die ich ab und zu auch mit dem alten C64 belege. Nun wollte ich eine möglichst komfortable Lösung schaffen, um ohne viel Hin-und-her-Steckerei alle diese Geräte mit dem Monitor stabil zu verbinden. Dabei sollte die Scart-Buchse jedoch erhalten bleiben (man weiß ja nie, wozu die noch gut sein kann!).

Erste Maßnahme war der Anschluß eines Verbindungskabels, an dessen einem Ende sich vier Cinch-Stecker (monitorseitig) und am anderen ein gewöhnlicher fünfpoliger DIN-Stecker befinden. Nach Studium des Anschlußplanes für den C64-Videoausgang ergab sich die Anschlußbelegung in Bild 1. Dieses Kabel kann man in jedem gut sortierten Radio-Fachgeschäft erhalten. An diesen Anschluß paßt übrigens auch der Ur-Heimcomputer VC20 von Commodore.

Am C64 heißen die Anschlüsse
1: Luminanz (unbenutzt)
2: GND
3: Audio out
4: FBAS
5: Audio in (unbenutzt)

An diesen DIN-Stecker wollte ich nun auch das Fernsehbild anschließen, da dieser Stecker am besten zugänglich ist. Dazu benötigt man wieder ein fünfpoliges DIN-Kabel, das diesmal jedoch eine DIN-Kupplung an einem Ende benötigt. Das andere Ende habe ich an einen Euro-AV-Stecker angelötet (so einer bleibt später vom Amiga-Kabel über). Die Belegung der Scart-Buchse kann man der Bedienungsanleitung für den Videorecorder entnehmen. Da sie genormt ist, paßt sie auch an jeden anderen. Für meine Zwecke brauchte ich folgende Pins:
1 + 3 Audio = Ton 1 und Ton 2 (Ton links und Ton rechts)
19 FBAS = zusammengesetztes Bildsignal
16 Schnelle Austastung (darauf kann man auch verzichten)

Damit erhält man eine Belegung wie in Bild 2.

Mit diesen beiden Kabeln kann man nun durch einfaches Umstecken den C64 und das Fernsehbild (übrigens in hervorragender Qualität) auf den Monitor bringen. Leider ist der Monitor nur mit einem Monoverstärker und einem Lautsprecher ausgerüstet. Beim 1084S kann man beide Kanäle nutzen. Der noch freie Stecker "Ton 2" bringt den zweiten Tonkanal bereits bis zum Monitor. Wenn

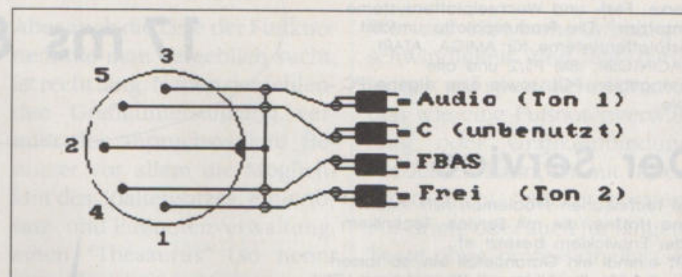


Bild 1. Anschluß Cinch auf DIN

ich diesen eines Tages auf Stereo umrüste, kann ich auch in Stereo fernsehen. Zur Zeit muß ich jedoch den Videorecorder auf Mono umstellen, um alle Töne zu hören.

Bei dieser Belegung mußte ich jedoch feststellen, daß ich kein vernünftiges Fernsehbild oder C64-Bild erhalte, wenn der Amiga eingeschaltet ist. Schuld daran ist, daß der Monitor jetzt zwei widersprüchliche Synchronisationssignale erhält (FBAS). Wenn man nun das Synchronsignal vom Amiga abschaltet, erhält man, nach Umschalten von RGB auf CVBS (= FBAS) im Bedienfeld des Monitors, ein befriedigendes Ergebnis. Zunächst habe ich mir also dadurch weitergeholfen, den Scart-Stecker aus dem Monitor zu ziehen. Das war mir jedoch zu umständlich. Darum habe ich, als ich eine neue Steckverbindung zum Amiga herstellte, gleich einen Schalter mit ins Monitorgehäuse eingebaut.

Die neue Steckverbindung besteht aus einer Joystick-Stecker-Stiftleiste (eine solche für Print-Montage ist am günstigsten), die an den Anschlüssen der Scart-Buchse festgelötet wurde. Vom Computer her habe ich den Scart-Stecker vom Kabel abgetrennt (auch das Audiokabel), indem ich mit einem Seitenschneider die Drähte hinter den Steckmessern abschnitt. Somit hatte ich den Stecker für den Anschluß am Videorecorder frei. Statt dieses Steckers muß nun eine Joystick-Kupplung (Buchsenleiste), die im Computerhandel erhältlich ist, angelötet werden.

Das Kabel muß folgendermaßen vorbereitet werden: Die Kunststoffummantelung wird auf zirka 10 cm entfernt. Die zum Vorschein kommende verseilte Abschirmung muß zunächst "entseilt" werden. Dabei läßt man die freigelegte unisolierte Litze zunächst unberührt, der Rest der Abschirmung kann am Ende der Kunststoffummantelung abgeschnitten werden. Auch die Metallfolie darunter sowie die zwischen den innenliegenden Leitungen befindlichen "Bindfäden" werden so abgeschnitten. Um später einen Kurzschluß zu vermeiden, wird die Masseleitung, die von der Abschirmung übrigbleibt, jetzt isoliert. Dazu kann man Isolierband oder, die elegantere Lösung, einen Schrumpfschlauch nehmen. Von allen Leitungsenden wird jetzt zirka 0,5 cm Isolierung abgezogen. Jedes Ende wird verdrillt und anschließend mit Lötzinn leicht vorverzinnt. Das erleichtert das spätere Anlöten. Ebenso wird mit allen anderen Kabelenden verfahren.

Nun wollen wir das Computerkabel fertigstellen. Wir finden vier Innenadern, die einzeln abgeschirmt sind, vor. Die Farblei-

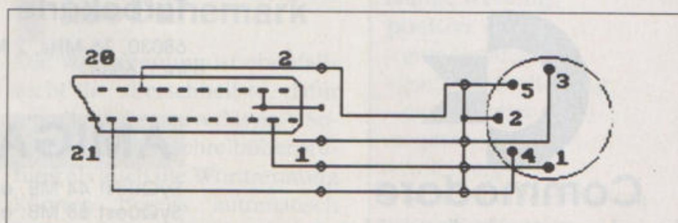


Bild 2. Anschluß DIN auf Scart

tungen sind mit den jeweiligen Farben (Rot, Grün, Blau) gekennzeichnet, wobei die schwarze Leitung Masse und die klare die eigentliche Farbleitung ist. Die vierte Ader besteht aus einer weißen Leitung (Synchronsignal = FBAS) und einer schwarz ummantelten Masseleitung (GND). Das Kabel ist leider zu stark, um in die Joystick-Stecker-Durchführung zu passen. Bei dem von mir verwendeten Kunststoffgehäuse brauchte ich jedoch nur mit einem Bohrer an einer Halbschale die Verstärkung des Führungsloches abzufräsen (3-mm-Bohrer auf Minibohrmaschine). Eine flexible Alternative stellt das Kabel eines ausgedienten Joysticks dar. Diese Kabel haben nur einen Nachteil: Die Pins 5, 7 und 9 sind ohne Anschluß. So kann man leider die Audioleitung nicht integrieren und mußte das externe Kabel weiterverwenden.

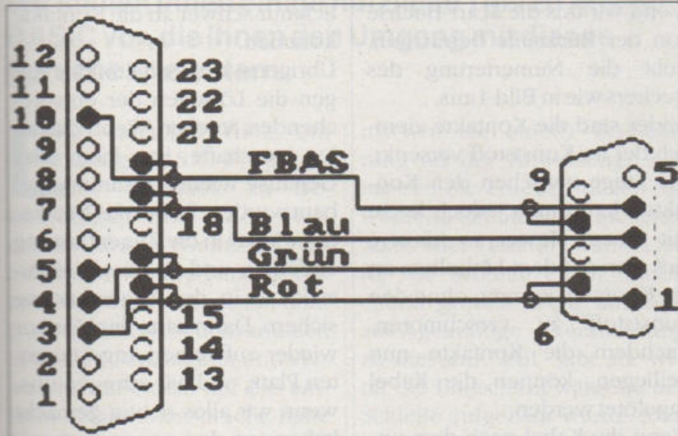


Bild 3. Scart-Anschluß im Eigenbau

Vorsicht Hochspannung!

Dieser Anschlußplan (Bild 3) ist fast identisch mit dem für Multisync-Monitore wie zum Beispiel den Commodore 1950. Nur die Masseleitung, die hier auf Pin 6 und 8 gelegt wurde, gehört auf Pin 9, den ich ja nicht verwenden konnte.

Kommen wir nun zum schwersten Teil der Arbeit: der Neuverdrahtung des Monitors. Da die Scart-Buchse erhalten bleiben soll, reicht es nicht, die Einbaubuchse aus- und dafür die neue einzubauen. Die Kabelenden müssen vielmehr an die durchzuführenden Buchsenkontakte (RGB, Sync, Audio, Masse) angelötet werden.

Wie kommen wir jetzt an diese Kontakte? Zunächst muß dazu das Monitorgehäuse geöffnet werden. Nachdem wir alle Anschlüsse vom Monitor entfernt haben (auch das Netzkabel aus der Steckdose ziehen!), stellen wir den Monitor auf den Bildschirmrahmen. Die jetzt nach oben weisende Anschlußleiste ist am abnehmbaren Gehäuseteil mit einer Schraube (neben den Cinch-Buchsen) befestigt. Da alle Anschlüsse auf der Hauptplatine angelötet sind, muß diese Schraube herausgedreht werden. Das Netzkabel liegt in Windungen in einer Zugentlastung. Daraus befreien wir es zunächst. Wenn wir die vier Schrauben entfernt haben, können wir die obere Gehäuseschale vorsichtig nach oben abziehen. Wenn ein Widerstand zu spüren ist, muß zunächst der Stecker für den Lautsprecher (das Kabel bildet die Bremse) von der jetzt schon erreichbaren Platine abgezogen werden. Dazu muß man den am Stecker befindlichen Hebel herunterdrücken, der die Verbindung verriegelt. Dann kann man das Gehäuse ganz abnehmen.

Der nächste Arbeitsschritt ist, das Innenleben (die Hauptplatine mit den Anschlüssen) des Monitors auszubauen. Dies ist erforderlich, um besser an die zu lötenden Stellen heranzukommen. Bevor wir damit beginnen, sollten wir uns genau merken, an welchen Stellen die Steckverbindungen später wieder angeschlossen werden müssen (ein kleiner Zettel, auf dem man die Verbindungen notiert, könnte hierbei recht nützlich sein).

An dem abgenommenen Gehäuseteil ist im Anschlußfeld eine Öffnung für einen weiteren RGB-Eingang vorgestanzt. Diese wird jetzt mit einem Bastelmesser, am besten von der Innenseite her, geöffnet (man kann die Öffnung auch auffräsen). Dies reicht jedoch noch nicht für die Joystick-Buchse. Wir müssen zusätzlich die nicht benötigte TTL-RGB-Buchse ausbauen und die beiden Löcher (elegant) durch einen Schnitt zu einer länglichen Öffnung erweitern. Durch dieses Loch wird jetzt das Audiokabel geführt, so daß die Enden im Monitor verlötet werden können (innen an den Pins der Scart-Buchse).

Bevor wir zum Anschließen kommen: Wenn der Monitor, wie bei mir, mit dem Schalter zur Synchronisationsabschaltung der Computerleitung versehen werden soll, ist dies jetzt zu tun. In das Gehäuseteil, das jetzt noch den Bildschirm umrahmt, habe ich an der rechten Seite ein Führungsloch für den Schalter (1x aus) aufgebohrt.

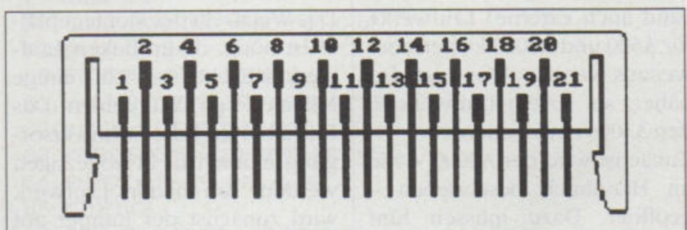


Bild 4. Scart-Pin-Numerierung

Wenn der Monitor wieder komplett vor uns steht: an der rechten Seitenwand, in Höhe des vorderen Bedienfeldes, direkt vor der Stoßnaht zwischen den Gehäusehälften. Diese Stelle gefiel mir am besten, da es die dem Umschalter CVBS/RGB am nächsten liegende Position ist und diese beiden Schalter allgemein zusammen umzuschalten sind. Ich habe einen Druckschalter in kleiner Bauform gewählt, der leicht an dieser Stelle unterzubringen ist. Solche Schalter erhält man in Elektronik-Bastelshops. An diesen Schalter habe ich zweiadrige Schaltlitze angelötet, die von der Länge her am Platinenträger entlang gut bis zur Scart-Buchse reicht. Am offenen Ende wird eine Hälfte einer doppelten Steckverbindung angelötet. Die Fortführung der Litze ab der Steckverbindung wird später an der einen Seite statt des FBAS-Drahtes an den, von der Platine getrennten, Scart-Anschluß und an der anderen Seite an den FBAS-Draht gelötet (der wird statt des Steckerpins an der Platine angelötet).

Doch nun zum Anschluß des RGB- und des Audiokabels am Scart-Stecker. Das erste Problem stellt sich dadurch, daß die Kontakte an der Rückseite des Steckers nicht zugänglich sind.

Ein Anschluß an der Platinenseite scheitert am Platzmangel für das Kabel. Also muß die Buchse ausgelötet werden! Wenn mit einem sehr feinen Lötkolben gearbeitet wird und man viel Fingerspitzengefühl aufbringt, geht es aber auch ohne das Auslöten. Mit einer Entlötpumpe ist das Entfernen des Lötzinns kein Problem. Wenn man über ein solches Gerät nicht verfügt, wird die Sache schon haarig. Man muß dann das Lötzinn stückweise mit dem Lötkolben abziehen. Man kann das Lötzinn aber auch von der Platine lösen, indem man es an eine dafür hergerichtete Drahtschlaufe bindet. Sind die Anschlüsse erst einmal befreit, kann man die Buchse durch Spreizen der Halteklammern leicht abziehen.

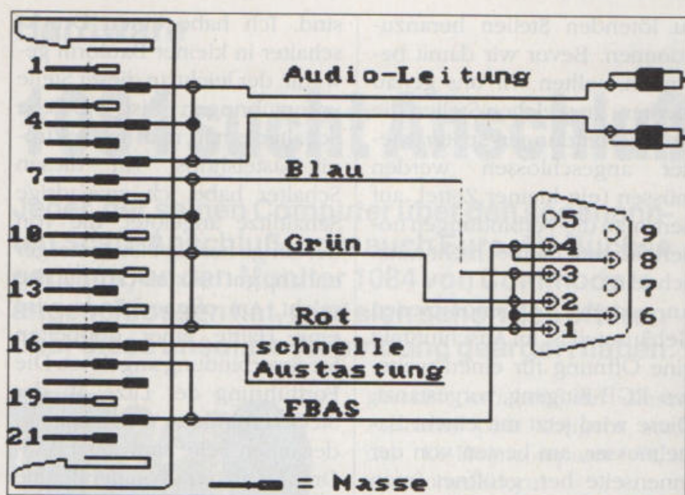


Bild 5. Der fertige Scart-Anschluß

Wie bereits oben erwähnt, habe ich eine Joystick-Kupplung für Print-Montage (übrigens eine direkt aus dem Amiga ausgebaute Joystick-Kupplung) verwendet. Da die Platine natürlich nicht mit den entsprechenden Leiterbahnen ausgestattet ist, habe ich die Halteklammern in eigens angebrachte Löcher in der Platine gesteckt und mittels eng angelegter Drahtwicklungen unter der Platine verlötet. Beim Bohren und Löten sollte man sehr vorsichtig vorgehen und den Kontakt zu den Leiterbahnen unbedingt vermeiden. Dies ist glücklicherweise nicht allzu schwer, da in diesem Bereich nur wenige Bahnen liegen. Die Befestigung hat sich als ausreichend stabil gezeigt, man könnte sie aber zum Beispiel durch Heißkleber noch besser fixieren. Die Kontakte wurden von mir von der Platine abgebogen und auf 5 mm gekürzt. An diese Kontakte wurden dann die Drähte, die mit der Scart-Buchse verbunden werden sollten, angelötet.

Bevor wir diese Arbeit erledigen können müssen wir aber noch, wie oben erwähnt, die TTL-RGB-Buchse auslöten.

Wenn wir uns die Scart-Buchse von der Rückseite betrachten, sieht die Numerierung des Steckers wie in Bild 4 aus.

Leider sind die Kontakte ziemlich tief im Kunststoff versenkt. Die Stege zwischen den Kontakten kann man jedoch leicht mit einem Messer so kürzen, daß man mit dem Lötkolben an die Kontakte kommt, ohne den Kunststoff zu verschmoren. Nachdem die Kontakte nun freiliegen, können die Kabel angelötet werden.

Wenn die Kabel nach dem unten angegebenen Anschlußschema angelötet sind, muß die Scart-Buchse wieder eingelötet werden. Dazu werden die Kontakte und Halteklammern zunächst in die vorgesehenen Löcher versenkt. Beim Verlöten sollte man darauf achten, die Kontakte nur möglichst kurz zu erhitzen, da sonst die Gefahr besteht, daß auch die Lötstellen an den Kabelenden schmelzen. Dann fängt man wieder von vorne an! An der bereits eingesetzten Buchse kann man nämlich nur schwer an die Kontakte kommen.

Übrigens: Alle Schaltbilder zeigen die Lötseiten der entsprechenden Stecker. Wenn das alles geschafft ist, kann das Gehäuse wieder zusammengebaut werden. Dabei das Netzkabel wieder in die Zugentlastung einfädeln und das neue Monitorkabel in der Zugentlastung sichern. Dann kann der Monitor wieder auf seinen angestammten Platz, und wir erfreuen uns, wenn wir alles richtig gemacht haben, an den neuen Features und einer sicheren Monitorverbindung zum Computer. ❑

Holger Wallmann (gg)

Einbautip

Internes Laufwerk

Für den A3000 wird ein zweites internes Laufwerk zu einem Preis von mehr als 200 Mark angeboten, interne (und auch externe) Laufwerke für A500 und A2000 kosten aber weitaus weniger. Was liegt also näher, als so ein Laufwerk in den A3000 einzubauen?

Zunächst wird der A3000 – wie im Handbuch beschrieben – geöffnet. Dazu müssen fünf Schrauben gelöst werden. Nach dem Abziehen des Gehäuses sind die Laufwerksschächte frei zugänglich.

Da das Disk-Buskabel zu kurz ist, kann man nicht einfach den leeren Stecker zum rechten Laufwerksschacht hinüberziehen. Es muß also anders vorgegangen werden. Dafür gibt es zwei Möglichkeiten:

1) Das bisherige interne Laufwerk (df0:) wird von der Montageplatte abgeschraubt, mit der bisher leeren Montageplatte verschraubt und im rechten Laufwerksschacht eingesetzt. Das Wechseln der Montageplatte ist nötig, da im linken Laufwerksschacht seitlich einige Millimeter an Platz fehlen. Das Buskabel und die Stromversorgung müssen nicht abgezogen werden. Am neuen Laufwerk wird zunächst der Jumper auf der Rückseite auf 1 gestellt, auf der Platine läßt sich diese Position ablesen. Anschließend wird das neue Laufwerk an das Montageblech angeschraubt (das bisher "df0:" gedient hat), dies erweist sich bei einem Chinon-Laufwerk als völlig problemlos. Danach wird die Stromversorgung aufgesteckt; hierbei kann nicht viel falschgemacht werden, da der Stecker verdrehsicher ist. Dann wird der bisher leere Stecker des Buskabels aufgesteckt. Dabei ist darauf zu achten, daß der rote Streifen des Kabel mit Pin 1 der Steckerleiste des Laufwerks auf einer Seite zu liegen kommt. Nun kann das Laufwerk im linken Schacht eingebaut werden – fertig.

2) Das Laufwerk (df0:) wird aus seinem Schacht genommen, das Buskabel wird abgezogen. Die Montageplatte und das Stromversorgungskabel brauchen nicht entfernt zu werden. An der Rückseite des Laufwerks wird der Jumper auf 1 umgesteckt (siehe oben). Nachdem der bisher leere Stecker des Buskabels aufgesteckt wurde (auf den roten Streifen achten!), kann das Laufwerk wieder im linken Schacht eingebaut werden. Am neuen Laufwerk wird zunächst der Jumper auf 0 gestellt und die Montageplatte angeschraubt. Danach werden die Stromversorgung (siehe oben) und das Buskabel aufgesteckt. Beim Buskabel ist wieder darauf zu achten, daß der rote Streifen mit Pin 1 der Laufwerkssteckerleiste auf einer Seite zu liegen kommt. Auskunft über Pin 1 findet sich auf der Platine des Laufwerks. Anschließend kann das Laufwerk im rechten Schacht eingebaut werden. Bevor der A3000 geschlossen werden kann, muß noch ein Jumper umgesteckt werden, um das neue Laufwerk im System anzumelden.

Dieser Jumper trägt die Bezeichnung "J351" und befindet sich auf der linken, frei zugänglichen Platine rechts oben von Denise, neben dem Videoslot. Dieser Jumper wird auf die Position 1-2 umgesteckt.

Anschließend wird aus dem A3000-Gehäuse die rechte Laufwerksabdeckung entfernt. Je nach Laufwerkstyp muß unter Umständen die daneben befindliche (gelbe) Plastikkappe für die Laufwerks-LED entfernt werden. Eventuell muß die Zuleitung der Laufwerks-LED mit Hilfe eines Lötkolbens verlängert werden. Dies hängt vom verwendeten Laufwerkstyp ab. ❑

Gerald Seidler (tb)

Tips und Tools

Tips zu Amiga-BASIC

Wir stellen Ihnen einige nützliche Tips zu Amiga-BASIC vor, die Ihnen den Umgang mit dieser Sprache erleichtern.

Ein einfacher Trick, um AmigaBASIC-Programme eventuell zu beschleunigen? Bitte schön: Nutzen Sie Integer-(Ganzzahl-)Variablen! Dieser Typ benötigt nur zwei statt der vier Byte bei üblichen Fließkommavariablen einfacher Genauigkeit und verwendet außerdem nur das einfache, Maschinensprache-nahe Binärformat. Jeglicher Zugriff auf diesen Typ ist damit um einiges schneller. Am bequemsten ist ein Aufruf von "DEFINT ..." am Anfang des Programms, was die Angabe von "%" bei den Variablennamen erspart. Gerade bei Spielen ist dies konsequent auszunutzen, da diese selten Rechnungen durchführen, die Nachkommastellen erfordern. ❑

Oliver Gumtau (vb)

Geschwindigkeit

Wenn Sie unsicher sind, ob die eine oder doch die andere Spielart eines Programmteils in AmigaBASIC schneller ist, nutzen Sie die unten aufgeführte Routine einfach zum Austesten. Betten Sie die Anweisung(en) einfach in die For-Next-Schleife ein, starten Sie die Messung, machen Sie eine(!) Veränderung an der Anweisung, und messen Sie erneut. Spielen Sie während der Messung nicht mit Maus oder Tastatur. Wählen Sie für die Schleife einen hinreichend großen Wert, um eine aussagekräftige Veränderung zu messen. "VBI" gibt an, wie oft der Bildschirm während der Schleife aufgebaut wurde. Man kann mit der Meßroutine zum Beispiel zu folgenden Ergebnissen kommen, die in zeitkritischen Programmteilen Anwendung finden sollten.

- GOSUB ist schneller als CALL/SUB,
- REMs kosten je nach Textlänge zusätzlich Zeit,
- Hexadezimalzahlen müssen vom BASIC zeitaufwendig umgerechnet werden (also vermeiden),
- ON GOTO sollte x IF-Abfragen auf kontinuierliche Wertfolgen ersetzen,
- oder die Verwendung von den Systemroutinen "Move" und "Text" ist schneller als die entsprechenden POKEs + PRINT.

Lassen Sie keine Unsicherheiten mehr zu, testen Sie alles aus! ❑

Oliver Gumtau (vb)

```
POKE 12578561&,PEEK(12578561&) AND 127
POKE 12577025&,0
POKE 12576769&,0
FOR i%=1 TO 5000:
 'Hier steht Ihre Routine
NEXT
Zeit&=256*PEEK(12577025&) OR PEEK(12576769&)
PRINT "Dauer der Schleife:"
PRINT Zeit&;"VBIs",Zeit&/50;"sec"
```

Toolbox für AmigaBASIC

Jeder AmigaBASIC-Programierer hat so seine kleinen Hilfsprogramme, teils selbst erdacht, teils aus der Rubrik "Tips und Tricks" abgetippt. In der Util-Box sind die wichtigsten Hilfsprogramme eingebaut. Ein absolutes Muß für alle Einsteiger. Ob Farbbestimmungsprogramm, Phonem-Code, GET-Berechnung, ein kleiner Linienmustereditor oder Umrechnung von Dezimal nach Hexadezimal – alles ist vorhanden.

Da das Programm "Hinweise" darüber gibt, wo im Amiga-BASIC-Handbuch die betreffenden Anweisungen oder Befehle stehen, können wir hier verzichten, näher darauf einzugehen.

Hinweis: Da alle Programme eigenständig sind, ist das Hauptmenü nur nur nach Beendigung eines Programmteils wieder in Betrieb. Das Programm benötigt auf Ihrer Boot-Diskette im Libs-Ordner die »translator .library« und im DEVS-Ordner die »narrator.device«. ❑

Wolfgang Serafin (vb)

```
REM
REM AmigaBASIC Tools: Util-Box
REM Autor: Wolfgang Serafin
REM (c) AMIGA DOS 1992
REM

SCREEN 2,640,256,2,2
WINDOW 2," (c) 1991 by Wolfgang Serafin",(0,0)-(631,185),0,2
farb
maske
hauptmenue:
pstop
x%=MOUSE(1)
y%=MOUSE(2)
IF x%>30 AND x%38 AND y%54 AND y%70 AND y%86 AND y%102 AND y%158 AND y%370 AND x%50 AND y%1 THEN
IF pcol%=0 THEN col%=2 ELSE col%=0
PAINT(x%,y%),col%,1
r%=15:hz=0
FOR i%=375 TO 525 STEP 10
IF POINT(i%,55)=2 THEN hz=hz+2^r%
r%=r%-1
NEXT i%
LINE (535,50)-(590,60),0,bf
posi 535,59
PRINT "=&H"+HEX$(hz)
xp%=INT((x%-370)/10)+370
xp2%=2*(INT((x%-370)/10))
FOR i%=0 TO 112 STEP 16
PSET (xp%+i%,83),col%
NEXT i%
FOR i%=370 TO 466 STEP 32
LINE(xp2%+i%,99)-(xp2%+i%+1,99),col%
NEXT i%
END IF
IF x%400 OR y%168 THEN muster
loesche
END SUB

SUB pcode STATIC
LOCATE 3,35:PRINT "Phonem-Code siehe Handbuch 9-249 und 9-218"
LOCATE 21,40:COLOR 2:PRINT "E";:COLOR 1:PRINT "nde"
```

```
77: phon:
78: FOR roll=0 TO 23
79: SCROLL(261,50)-(619,150),0,-1
80: NEXT roll
81: LOCATE 17,40:INPUT "Eingabe";a$
82: LOCATE 18,40:PRINT "Ausgabe"
83: b$=TRANSLATE$(a$)
84: LOCATE 18,48:PRINT b$:SAY b$
85: c$=UCASE$(a$)
86: IF c$"E"THEN phon
87: loesche
88: END SUB
89:
90: SUB farben STATIC
91: PALETTE 3,0,0,0
92: LOCATE 3,37:PRINT "Palette-Anweisung siehe Hand-
buch 9-175"
93: LOCATE 6,45:PRINT "R"
94: LOCATE 8,45:PRINT "G"
95: LOCATE 10,45:PRINT "B"
96: LINE(350,100)-(550,135),3,bf
97: FOR i%=38 TO 70 STEP 16
98: quad 369,i%,531,i%+10,1
99: LINE(370,i%+1)-(380,i%+9),2,bf
100: NEXT i%
101: LOCATE 21,40:PRINT "Hauptmenue"
102: quad 300,158,400,168,1
103: e%=0
104: col:
105: pstop
106: xm%=MOUSE(1)
107: ym%=MOUSE(2)
108: IF xm%>370 AND xm%38 AND ym%38 AND ym%54
AND ym%70 AND ym%400 OR ym%168 THEN col
120: loesche
121: PALETTE 3,0,0,1
122: END SUB
123:
124: SUB schieber (fa%,ly%,yp%,xm%,e%) STATIC
125: IF e%=0 THEN
126: e%=1
127: FOR f%=0 TO 2
128: ft(f%)=0
129: NEXT f%
130: END IF
131: r%=INT(xm%-375)/10
132: rp%=370+(r%*10)
133: ft(fa%)=r%/15
134: LINE(370,yp%)-(530,yp%+8),0,bf
135: LINE(rp%,yp%)-(rp%+10,yp%+8),2,bf
136: LOCATE ly%,69:PRINT USING"#.##";ft(fa%)
137: PALETTE 3,ft(0),ft(1),ft(2)
138: END SUB
139: SUB dezhex STATIC
140: LOCATE 3,42:PRINT "Dezimal in Hexadezimal"
141: LOCATE 20,40:PRINT " Neue Berechnung (j/n)"
142: neue:
143: FOR roll=0 TO 23
144: SCROLL(261,50)-(619,150),0,-1
145: NEXT roll
146: LOCATE 17,40:INPUT "Dezimal";a
147: LOCATE 18,43:PRINT "Hexa &H"+HEX$(a)
148: h:a$=INKEY$ :IF a$="" THEN h
149: IF a$ ="j" THEN neue
150: loesche
151: END SUB
152:
153: SUB getberech STATIC
154: neu:
155: loesche
156: LOCATE 3,36:PRINT "GET-Berechnung siehe auch
Handbuch 9-84"
157: LOCATE 8,40:INPUT" x1 ";x1
158: LOCATE 9,40:INPUT" y1 ";y1
159: LOCATE 10,40: INPUT" x2 ";x2
160: LOCATE 11,40: INPUT" y2 ";y2
161: LOCATE 12,40: INPUT"Tiefe";tiefe
162: byte=6+(y2-y1+1)*2*INT((x2-x1+16)/16)*tiefe
163: LOCATE 14,40:PRINT "Es werden";byte;"Bytes benoe-
tigt"
164: LOCATE 16,40: PRINT "Das ergibt";byte/2;"Elemente."
165: LOCATE 18,40:PRINT " Neue Berechnung (j/n)"
166: halt:a$=INKEY$ :IF a$="" THEN halt
167: IF a$ ="j" THEN neu
168: loesche
169: END SUB
170:
171: SUB quad (x1%,y1%,x2%,y2%,col%) STATIC
172: LINE(x1%,y1%)-(x2%,y2%),col%,b
173: LINE(x1%-1,y1%)-(x1%-1,y2%),col%
174: LINE(x2%+1,y1%)-(x2%+1,y2%),col%
175: END SUB
176:
177: SUB farb STATIC
178: PALETTE 0,0,0,0
179: PALETTE 1,1,0,0
180: PALETTE 2,0,1,0
181: PALETTE 3,0,0,1
182: END SUB
183:
184: SUB posi (xpos%,ypos%) STATIC
185: POKEW WINDOW(8)+36,xpos%
186: POKEW WINDOW(8)+38,ypos%
187: END SUB
188:
189: SUB pstop STATIC
190: WHILE MOUSE(0)0:WEND
191: WHILE MOUSE(0)=0:WEND
192: END SUB
193:
194: SUB loesche STATIC
195: LINE (261,11)-(619,179),0,bf
196: END SUB
197:
198: SUB maske STATIC
199:
200: quad 260,10,620,180,1
201: quad 10,10,200,180,1
202: LOCATE 3,10:COLOR 2:PRINT "UTIL-BOX"
203: quad 30,38,180,48,1
204: LOCATE 6,7:COLOR 1:PRINT "GET-Berechnung"
205: quad 30,54,180,64,1
206: LOCATE 8,8:PRINT "Linien-Muster"
207: quad 30,70,180,80,1
208: LOCATE 10,11:PRINT "Palette"
209: quad 30,86,180,96,1
210: LOCATE 12,9:PRINT "Phonem-Code"
211: quad 30,102,180,112,1
212: LOCATE 14,10:PRINT "Dezi>Hexa"
213: quad 30,158,180,168,1
214: LOCATE 21,12:PRINT "Quit"
215: END SUB
```

Cursor-Vielfalt

Warum immer dieser einheitliche Cursor bei den String-(Integer-) Gadgets? Es geht doch auch anders. Zum Beispiel, wenn das Gadget-Flag auf "GadgHBox" anstatt auf "GadgHComp" gesetzt wird ($1 statt $0). Dies zeigen die folgenden Listings. ❑

Hanns Holger Rutz (tb)

»BoxStringGG.asm« wurde mit dem Assembler »Asm-One 1.01« erstellt.

```
Main      move.l   4.w,a6             ; ExecBase
          lea      IntName(pc),a1     ; LibraryName
          jsr      -408(a6)           ; OldOpenLibrary ()
          move.l   d0,a6              ; IntuiBase
          lea      NewWindow(pc),a0   ; NewWindow-Struktur
          jsr      -204(a6)           ; OpenWindow ()
          move.l   d0,a5              ; Window
          move.l   a6,a4              ; IntuiBase
          move.l   4.w,a6             ; ExecBase
          move.l   86(a5),a3          ; UserPort
          move.l   a3,a0              ; Port
          jsr      -384(a6)           ; WaitPort ()
          move.l   a3,a0              ; Port
          jsr      -372(a6)           ; GetMsg ()
          move.l   d0,a1              ; Message
          jsr      -378(a6)           ; ReplyMsg ()
          move.l   a4,a6              ; IntuiBase
          move.l   a5,a0              ; Window
          clr.l    62(a0)             ; FirstGadget
          jsr      -72(a6)            ; CloseWindow ()
          move.l   a6,a1              ; Library
          move.l   4.w,a6             ; ExecBase
          jsr      -414(a6)           ; CloseLibrary ()
          moveq.l  #0,d0              ; CLI Fehlercode
          rts                         ; Tschüß
IntName   dc.b     "intuition.library",0
Title     dc.b     "StringGadgetDemo mit 'GadgHBox' by H.H.Rutz",0
Buffer    dcb.b    52,0
Undo      dcb.b    52,0
          even
NewWindow dc.w     0,0,640,30
          dc.b     0,1
          dc.l     $200,$2!$4!$8!$1000!$10000,BoxString,0,Title,0,0
          dc.w     640,30,640,30,$1
BoxString dc.l     0                  ; Zeiger auf nächstes Gadget
          dc.w     20,16              ; Linke und obere Ecke
          dc.w     400,8              ; Breite und Höhe
          dc.w     $1                 ; GadgetFlags (GadgHBox)
          dc.w     $2,$4              ; ActivationFlag und GGTyp
dc.l      0,0,0,0  ; GGRend,SelRend,Text,MutEx
          dc.l     Special            ; SpecialInfoBlock
          dc.w     1,0,0              ; GadgetNummer,-(User-Daten)
Special   dc.l     Buffer,Undo        ; Puffer für den Text
          dc.w     0,50,0,0           ; BufPos,MaxChar,DisPos,UndoPos
          dc.w     0,0,0,0            ; NumChar,DisCount,Left/TopEdge
          dc.l     0,0,0              ; Layer,LongInt,KeyMap
```

Das gleiche Beispiel nun in AmigaBASIC umgesetzt:
»BoxStringGG.bas«

```
DECLARE FUNCTION AllocMem& LIBRARY
DECLARE FUNCTION GetMsg& LIBRARY
LIBRARY ":bmaps/exec.library"
DECLARE FUNCTION OpenWindow& LIBRARY
LIBRARY ":bmaps/intuition.library" Main:
   Memory& = AllocMem& (128&,65538&)' Speicher Chip!Clear
   Title$ = "StringGadgetDemo mit 'GadgHBox' by H.H.Rutz" + CHR$ (0)
   Buffer$ = STRING$(52,0)
   Undo$ = STRING$(52,0)
   POKEW Memory& + 4 , 640                 ' FensterBreite
   POKEW Memory& + 6 , 30                  ' FensterHoehe
   POKE Memory& + 9 , 1                    ' BlockPen
   POKEL Memory& +10 , 512                 ' IDCMP-Flags (CloseWindow)
   POKEL Memory& +14 , 69646&              ' Flags
   POKEL Memory& +18 , Memory& +48         ' Zeiger auf das StringGG
   POKEL Memory& +26 , SADD (Title$)       ' Title des Fensters
   POKEW Memory& +46 , 1                   ' Screentyp (WB)
   POKEW Memory& +48 + 4 , 20              ' Linke Ecke
   POKEW Memory& +48 + 6 , 16              ' Obere Ecke
   POKEW Memory& +48 + 8 , 400             ' Breite
   POKEW Memory& +48 +10 , 8               ' Hoehe
   POKEW Memory& +48 +12 , 1               ' Flags (GadgHBox)
   POKEW Memory& +48 +14 , 2               ' Activation-Flags
   POKEW Memory& +48 +16 , 4               ' Gadgettyp (String)
   POKEL Memory& +48 +34 , Memory& +92     ' SpecialInfo
   POKEW Memory& +48 +38 , 1               ' GadgetNummer
   POKEL Memory& +92 + 0 , SADD (Buffer$)  ' Zeiger auf die Text-
   POKEL Memory& +92 + 4 , SADD (Undo$)    ' puffer
   POKEW Memory& +92 +10 , 50              ' Anzahl der max. Zeichen
   Win& = OpenWindow& (Memory&)            ' Fenster oeffnen
   IF Win& = 0 THEN GOTO FreeMemory        ' Prg beenden wenn Fehler
   UserPort& = PEEKL (Win& + 86)           ' Selbsterklaerend...
   CALL WaitPort& (UserPort&)              ' Auf Nachricht warten
   Msg& = GetMsg& (UserPort&)              ' Nachricht abholen
   CALL ReplyMsg& (Msg&)                   ' und beantworten
   POKEL Win& +62 , 0                      ' Gadgets entfernen
   CALL CloseWindow& (Win&)                ' Fenster schliessen
   CALL FreeMem& (Memory&,128&)            ' Speicher freigeben
   LIBRARY CLOSE                           ' Libraries schliessen
END                                        ' Ende
```

Bitoperationen für Hochsprachen

Um ein Bit zu "toggeln", das heißt in den jeweils anderen Zustand zu setzen, braucht man keine aufwendigen Abfragen, mit denen immer nur ein Bit "getoggelt" werden kann. Dazu müssen die Bits der zu verändernden Variablen "Exklusiv-Oder" (logisches "Entweder-Oder") verknüpft werden.

Ein Beispiel:
Bit 1 = 0011
Bit 2 = 0101
XOR = 0110

Bei gleichen Bits ist das Ergebnis immer "FALSCH", also 0. Erst wenn die zu vergleichenden (verknüpfenden) Bits ungleich sind, ist das Ergebnis "WAHR" und entspricht damit 1.

```
variable = variable ^ bitmaske;
```

Mit Hilfe der Exklusiv-Oder-Operation kann man auch die Inhalte von zwei Variablen ohne Hilfsvariable vertauschen: Als erstes Exklusiv-Oder-verknüpft man Variable 1 mit Variable 2 und speichert das Ergebnis in Variable 1. Dann Exklusiv-Oder-verknüpft man Variable 2 mit der neuen Variable 1 in Variable 2, die jetzt den Anfangswert von Variable 1 enthält. Als letzten Schritt muß jetzt Variable 1 das Ergebnis einer dritten XOR-Operation, nämlich der Anfangswert von Variable 2, zugewiesen werden. Wieder ein C-Beispiel:

```
UBYTE var1,var2;
...
var1 = var1 ^ var2;
var2 = var1 ^ var2;
var1 = var1 ^ var2;
```

Man muß aber darauf achten, daß beide Variablen vom gleichen Typ sind, das heißt, die gleiche Größe haben. ❑

Stefan Berendes (tb)

Strategiespiel

Orientierungsprobleme?

Länder auf Landkarten sind in den seltensten Fällen rechteckig. Wie kann der Programmierer eines Spiels dafür sorgen, daß der Spieler solche unregelmäßige Flächen mit der Maus anklicken kann?

Bei der Umsetzung eines Strategiespiels (siehe Kurs in AMIGA DOS 9'91 und 10'91) benötigt man oft Landkarten (oder ähnliches). Das Problem ist die Programmierung der Abfrageroutine, die auswertet, welches Land der Spieler selektiert hat. Hier einige Ideen:

– Die einzelnen Länder werden mit unterschiedlichen Farben gefüllt. Die Abfrage ist dann zum Beispiel:

"land% = point(mausx%,mausy%)"

Dafür dürfte es schwierig sein, die Länder grafisch ansprechend zu gestalten.

– Die Länder werden in Raster (zum Beispiel: 8x8) aufgeteilt. Die Abfrage ist dann

"land% = feld%(fix(mausx%/8),fix(mausy%/8))"

Der Nachteil ist, daß die Länder "unnatürlich" eckig aussehen.

– Die Länder werden mit den Cursortasten oder einem Joystick ausgewählt, indem über dem jeweiligem Land ein Pfeil oder ähnliches erscheint. Diese Methode ist meiner Meinung nach dem Amiga aber nicht angemessen.

Eine Abfrage auf jeden einzelnen Punkt des Bildschirms resultiert mit Sicherheit in einer wahren IF-THEN-Flut, die auch dem geübtesten Programmierer Tränen in die Augen treiben. Trotzdem gibt es eine Methode, um unregelmäßige Flächen (wie in Bild 1 zu sehen) "auf das Pixel genau" mit der Maus anzuklicken. Zu diesem Zweck muß der Programmierer eine Maske der entsprechenden Fläche erstellen.

Eine Lösung

Da ja die Grafik in der Regel mit einem Malprogramm erstellt wurde, kann genau dieses auch dazu benutzt werden, um eine Maske (siehe Bild 2) der entsprechenden Landkarte zu erstellen.

Für die Auswertung der einzelnen Mausklicks wird nun statt der Karte die Maske derselben verwendet.

Offset	Länge	Inhalt
$00	LONG	Zeiger auf die zweifarbige Grafik von Land 1 oder 0 (= kein Land mehr)
$44	WORD	X-Koordinate, wo die zweifarbige Grafik auf der Karte liegt (vom linken Rand); zum Beispiel 22 (Abbildung)
$66	WORD	Y-Koordinate, wo die zweifarbige Grafik auf der Karte liegt (vom oberen Rand); zum Beispiel 44 (Abbildung)
$88	WORD	Breite der zweifarbigen Grafik; sie muß durch 8 teilbar sein, da sie später in Bytes umgerechnet wird (z.B. 48)
$a10	WORD	Höhe der zweifarbigen Grafik; zum Beispiel 61 (Abb.)
$c12	LONG	Zeiger auf die zweifarbige Grafik von Land 2 oder 0 ...

Tabelle1. Zeiger für "GetCountry" (A0)

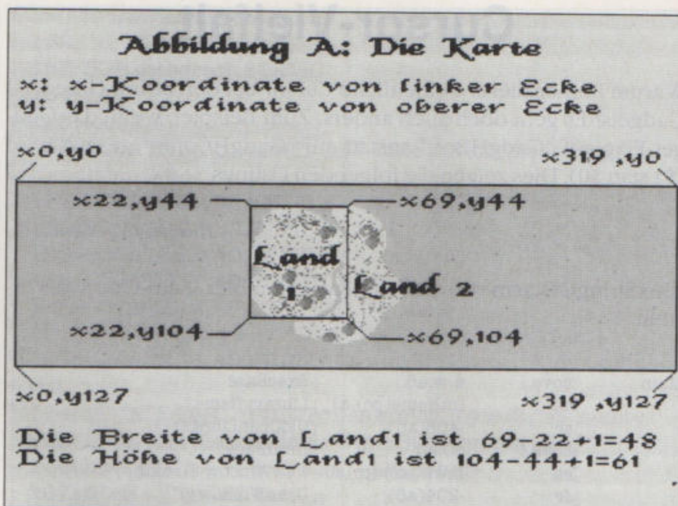


Bild 1. Die ursprüngliche Landkarte ...

Als Rückgabewert von "GetCountry" enthält "d0" die Nummer des Landes (LONG) oder 0, wenn kein Land selektiert wurde. Hier aber nun die Routine:

```
GetCountry  movem.l  d1-d6/a0/a1,-(sp)   ; Benutzte Register speichern
            moveq.l  #0,d0               ; Land-Nummer auf 0 setzen
.loop       move.l   (a0)+,a1            ; Nächster Grafik Pointer
            cmpa.l   #0,a1               ; Ende erreicht?
            bne.s    .cont               ; Naa, branch
            moveq.l  #0,d0               ; Land-Nummer = 0 (Fehler)
            bra.s    .quit               ; Sub verlassen
.cont       addq.l   #1,d0               ; Land-Nummer erhöhen
            movem.w  (a0)+,d3/d4/d5/d6   ; X-Offs,Y-Offs,Breite und Höhe
            movem.w  d1/d2,-(sp)         ; Alte X/Y-Koordinaten sichern
            sub.w    d3,d1               ; X-Offset vom Grafikrand
            blt.s    .next               ; Zu niedrig? Branch
            cmp.w    d1,d5               ; Zu hoch?
            ble.s    .next               ; - Branch
            sub.w    d4,d2               ; Y-Offset vom Grafikrand
            blt.s    .next               ; Zu niedrig? Branch
            cmp.w    d2,d6               ; Zu hoch?
            ble.s    .next               ; - Branch
            lsr.w    #3,d5               ; Grafikbreite in Bytes
            mulu.w   d2,d5               ; Mit Y-Offset multiplizieren
            move.w   d1,d2               ; X-Offset > d2
            lsr.w    #3,d1               ; X-Offset in Bytes
            add.w    d1,d5               ; Zum Offset addieren
            moveq.l  #7,d1               ; Zu testendes
            and.b    #7,d2               ; Bit
            sub.b    d2,d1               ; errechnen
            btst     d1,(a1,d5.w)        ; Und testen
            beq.s    .next               ; Nicht gesetzt, weitersuchen
            addq.l   #4,sp               ; Stackadresse korrigieren
.quit       movem.l  (sp)+,d1-d6/a0/a1   ; Benutzte Register zurücksetzen
            rts                          ; Und sub verlassen
.next       movem.w  (sp)+,d1/d2         ; Alte X/Y-Koordinaten laden
            bra.s    .loop               ; Und Prozedur wiederholen
```

– Außer der Karte, die der Spieler zu sehen bekommt, liegen die einzelnen Länder als zweifarbige Grafiken einzeln noch einmal im Speicher, wobei dort ein gesetztes Bit (beziehungsweise ein gesetzter Bildpunkt) bedeutet, daß das Land auf dieser Position selektiert wurde.

Eine Beispielkarte (siehe Bild 1 und Bild 2) zeigt die zweifarbige Grafik von Land 1. Um die Abfrage zu erleichtern, habe ich Abfrageroutinen in AmigaBASIC (1.2) und Assembler (Asm-One 1.01) geschrieben.

Assembler:

Die Subroutine "GetCountry" wird mit "bsr/jsr" aufgerufen, wobei folgende Parameter übergeben werden müssen:

d1 = Maus-X-Koordinate in Pixel von der linken Ecke der Karte (WORD);

d2 = Maus-Y-Koordinate in Pixel von der oberen Ecke der Karte (WORD);

a0 = Zeiger auf eine wie folgt aufgebaute Struktur (LONG) (siehe Tabelle 1).

AmigaBASIC:

Das Sub "GetCountry" wird mit

"CALL GetCountry (num%, mausx%,mausy%,tabe$)"

aufgerufen. Die Parameter haben folgende Bedeutung:

num% = enthält nach dem Aufruf von "GetCountry" die Nummer des Landes oder eine 0, wenn keines selektiert wurde.

mausx% = die Maus-X-Koordinate in Pixel (=Bildpunkte) von der linken Ecke der Karte.

mausy% = die Maus-Y-Koordinate in Pixel von der oberen Ecke der Karte.

tab$ = ein String, der sich wie folgt zusammensetzt:

tab$ = mkl$(Grafik&)+mki$(xOffs%)+mki$(yOffs%)+mki$(Breite%)+mki$(Hoehe%)

Grafik& = Zeiger auf die zweifarbige Grafik für Land 1; enthält man zum Beispiel mit "Grafik& = AllocMem& (Size&, Reqs&)" oder "Grafik& = SADD (Land1$)".

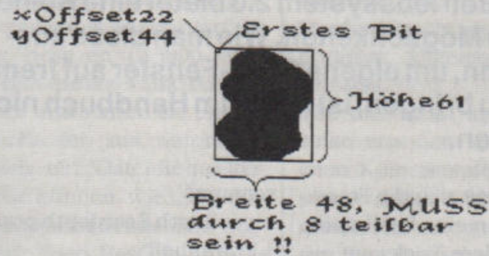


Bild 2. ... und die Maske für die Auswertung

xOffs% = X-Koordinate, wo die Grafik auf der Karte liegt (vom linken Rand); zum Beispiel: 22.

yOffs% = Y-Koordinate, wo die Grafik auf der Karte liegt (vom oberen Rand); zum Beispiel: 44.

Breite% = Breite der zweifarbigen Grafik; muß durch 8 teilbar sein, da sie später in Bytes (8 Pixel/Bit) umgerechnet wird; zum Beispiel: 48.

Hoehe% = Höhe der zweifarbigen Grafik; zum Beispiel: 61 (siehe Abbildung).

Danach folgt:

tab$ = tab$ + mkl$(Grafik2&)+mki$(xOffs2%)... (Grafik2,3,4...)

oder, wenn alle Länder in der Tabelle aufgenommen worden sind:

tab$ = tab$ + mkl$(0)

Das Unterprogramm (Sub):

Schlußbemerkung: Die Angaben, wie X-Offset und ähnliches, beziehen sich (damit man nachfolgen kann) auf die Abbildungen, die hoffentlich verständlich sind. Die zweifarbigen Grafiken und die Tabellen können sich sowohl im Chip- als auch im Fast-Memory befinden. Der Speicher kann also mit

"Speicher&=AllocMem&(Size&,1&)"

angefordert werden (bei Strings ist das egal). ❑

```
SUB GetCountry (num%,mx%,my%,tabelle$) STATIC
        num% = 1                                          'Land 1
GCLoop: adr& = CVL(MID$(tabelle$,num%*12-11,4))           'Grafikadresse
        IF adr& = 0 THEN num% = 0 : EXIT SUB              'Kein Land? - raus
        xoff% = mx% - CVI(MID$(tabelle$,num%*12-7,2))     'xOffset
        yoff% = my% - CVI(MID$(tabelle$,num%*12-5,2))     'yOffset
        xsize% = CVI(MID$(tabelle$,num%*12-3,2))          'GrafikBreite
        ysize% = CVI(MID$(tabelle$,num%*12-1,2))          'GrafikHoehe
        IF xoff%<0 OR xoff%=>xsize% OR yoff%<0 OR yoff%=>yend% THEN
         num% = num% + 1: GOTO GCLoop                     'Naechstes Land
        END IF
        bit% = 2^(7-(xoff% AND 7))                        'zu testendes Bit
        offs! = xsize%/8*yoff%+FIX(xoff%/8)               'zu testendes Byte
        IF (PEEK(adr&+offs!) AND bit%) = 0 THEN num% = num% +1: GOTO
        GCLoop
END SUB
```

Hanns Holger Rutz (tb)

Shells auf Public-Screens

Shell auf Abwegen

Das neue Betriebssystem 2.0 bietet eine Reihe von neuen Möglichkeiten. Wie man diese ausnutzen kann, um eigene Shell-Fenster auf fremde Screens zu bringen, ist leider im Handbuch nicht beschrieben.

Dabei stellt sich die Frage, wie man eine Shell ohne besondere Tricks auf eigene Screens schaltet. Bekanntlich unterstützt 2.0 sogenannte "Public-Screens", also Screens, die nicht vom Betriebssystem verwaltet werden, aber verschiedenen Programmen zugänglich sind. Eine komfortable Lösung für alle, die an mit Shell-Fenstern überfüllten Workbench-Screens leiden, ist es, diese auf eigenen "Public-Screens" zu öffnen, der CON-Handler im 2.0-ROM unterstützt das. Zur Verwaltung der Screens wird das Programm »PSX« von Steve Tibbet auf Fish-Disk #483 benötigt. Ich habe dieses Script-Paket »SShell« für "Screen-based Shell" genannt. Wie aus den obigen Erklärungen ersichtlich ist, läuft es nur ab AmigaOS-Version 2.0.

Installation

Die folgenden Scripts müssen unter den angegebenen Namen nach "S:". Bei allen sollte mittels "Protect" das S-Bit gesetzt werden. Die Zeilennummern dienen nur der Orientierung und dürfen nicht mit abgetippt werden.

Datei "NewSShell":

```
1 .KEY dpth
2 .BRA [
3 .KET ]
4 set tempscrnum $myscrnum
5 set myscrnum $scrcnt
6 setenv scrcnt `eval $scrcnt + 1`
7 setenv sshellcnt `eval $sshellcnt+ 1`
8 set tempdepth $scrdepth
9 set scrdepth [dpth$2]
10 psx open "$sshelltitle$myscrnum"
hires depth $scrdepth pop
11 newshell "con:0/0/640/256/AmigaShell/
CLOSE/BACKDROP/NOBORDER/NOSIZE/
SCREEN$sshelltitle$myscrnum" from
s:SShell-Startup
12 set scrdepth $tempdepth
13 unset tempdepth
14 set myscrnum $tempscrnum
15 unset tempscrnum
```

Zeile 10 kann um den Parameter LACE für einen Interlaced-Screen ergänzt werden. Zeile 11 in ihrer obigen Version ist für eine Auflösung von 640x256 Punkten ausgelegt. Sie muß gegebenenfalls an andere Display-Modi angepaßt werden.

Datei "Shell-Startup" (vorher das alte Shell-Startup in "Sh-Startup" umbenennen):

```
1 echo "*Ec" noline
2 set ownscr FALSE
3 set scrdepth $wbdepth
4 set myscrnum 0
5 unset tempscrnum
6 unset tempdepth
7 execute s:Sh-Startup
```

Die Datei "Sh-Startup" (die alte Shell-Startup-Datei) muß gegebenenfalls daran angepaßt werden, daß es nicht mehr von "NewShell", sondern von "Execute" ausgeführt wird (.KEY-, .BRA-, .KET-Zeilen einfügen).

Datei "SShell-Startup":

```
1 echo "*Ec*E[4y"
2 set ownscr TRUE
3 unset tempdepth
4 unset tempscrnum
5 alias EndCLI echo "EndCLI: Please use
»EndSh« instead."
6 alias EndShell echo "EndShell: Please
use »EndSh« instead."
7 execute s:Sh-Startup
```

Datei "EndSh":

```
1 unalias endcli
2 if $ownscr eq TRUE
3 run >nil: wait 1 secs +
4 execute s:SShell-Cleanup
5 endif
6 endcli
```

Datei "SShell-Cleanup":

```
1 psx close
"$sshelltitle$myscrnum"
2 setenv sshellcnt `eval
$sshellcnt - 1`
3 if $sshellcnt eq 0
4 setenv scrcnt 1
5 endif
```

Datei "ChangeSShellTitle":

```
1 .KEY NewTitle
2 .BRA {
3 .KET }
4 if not $sshellcnt eq 0
5 echo "You first have to close all
SShell screens. Aborting."
6 quit 5
7 endif
8 setenv sshelltitle "{newtitle}"
```

Das Programm »PSX« muß irgendwo in den Suchpfad. Einige Environment-Variablen werden benötigt. Nach dem Booten müssen sie gewisse Ausgangswerte haben, daher sollte man folgende Dateien mit dem angegebenen Inhalt in "EnvArc:" haben. Die weitere Verwaltung übernimmt »SShell«.

scrcnt "1"
sshellcnt "0"
sshelltitle "SShellScreen"
wbdepth "2" (enthält Tiefe des Workbench-Screens, gegebenenfalls anpassen)

Bedienung von »SShell«

Eine »SShell« wird mit dem Befehl "NewSShell" gestartet. Der einzig mögliche und dazu noch optionale Parameter ist eine Zahl zwischen 1 und 4, der die Tiefe (Bitplane-Anzahl) des Shell-Screens angibt (2, 4, 8 oder 16 Farben). 4 ist micht zu empfehlen, da sich sowieso nur 8 Farben durch ANSI-Codes ansprechen lassen und der Chip- und Slow-RAM-Zugriff verlangsamt wird. Default ist "wbdepth".

Eine »SShell« muß mit "EndSh" beendet werden, damit auch der Screen automatisch geschlossen wird. Via "Alias" ist der Zugriff auf "EndCLI/EndShell" verwehrt, "Ctrl-\" funktioniert leider immer noch. Damit man es sich zur Gewohnheit machen kann, immer "EndSh" zu benutzen, unterstützt es auch normale Shells.

Nach der Installation von »SShell« enthält jede Shell die folgenden lokalen Variablen. Die ersten beiden dürfen nicht geändert werden, da "EndSh" sie benötigt.

ownscr: TRUE für eine »SShell«, FALSE für eine normale Shell
myscrnum: Nummer des »SShell«-Screens (nur gültig falls ownscr=TRUE)
scrdepth: Tiefe des Screens (enthält den Wert von "wbdepth" für normale Shells)

In der Environment-Variable "sshelltitle" ist der Standardtitel für »SShell«-Screens abgelegt. Diese Variable darf nur geändert werden, wenn keine »SShell« läuft (sshellcnt = 0). Ein geeignetes Zugriffsprotokoll stellt das Script "ChangeSShell-Title" dar. Für eine dauerhafte Änderung muß noch der Wert in "EnvArc:sshelltitle" geändert werden.

Dieser Titeltext wird noch um eine für jede »SShell« individuelle Zahl ergänzt, die in "myscrnum" abgelegt ist. Die Zahlen steigen so lange an, bis die letzte »SShell« mittels "EndSh" beendet ist. Dann geht es wieder von vorne los. ❑

Arno Eigenwillig (gg)

Haben Sie Tips zum neuen Betriebssystem? Dann nichts wie her damit. Wir suchen ständig für unsere Rubrik Tips zur Shell oder zur Workbench 2.0. Adresse:
Redaktion AMIGA DOS
Kennwort OS 2.0
Postfach 250
3440 Eschwege

PowerPacker und ARexx

Packen mit Auswahl

Wer »PowerPacker« häufig benutzt, wird wissen, daß es bei kleinen Dateien mitunter effektiver sein kann, eine Datei mit der Einstellung "VeryGood" oder "Good" anstatt "Best" zu packen – das Entpacken dauert wegen der geringeren Dichte logischerweise nicht so lange.

Die Gründe dafür sind »PowerPacker«-intern zu suchen und sollen hier nicht weiter besprochen werden. Es geht mehr um eine komfortable Lösung des Problems.

Um mit dem »PowerPacker« garantiert das beste Ergebnis zu erzielen, ist man also gezwungen, alle drei Varianten zu testen, um am Ende das kürzeste Ergebnis zu behalten. Genau diesen Weg schlägt auch das ARexx-Script ein (Scripts unterscheiden sich durch Programme in der Ausführung, ähnlich wie ein Interpreter (AmigaBASIC) benötigen sie die Programmierumgebung – in diesem Falle »ARexx«), das nicht nur aufgrund seines praktischen Nutzens, sondern auch als Beispielanwendung für Nicht-»PowerPacker«-Besitzer von Interesse sein dürfte.

Kopieren Sie das Script einfach unter dem Namen »PPTry.pprx« ins Verzeichnis »Rexx:« (so Sie ARexx besitzen (A3000- und A500-plus-Besitzer sind da ja im Vorteil), und starten Sie es vom »PowerPacker« aus. Sofort öffnet sich ein Datei-Requester, und Sie können wie gewohnt das zu bearbeitende File auswählen. Nach Bestätigung des Requesters wird die Datei dreimal gepackt und das jeweilige Ergebnis im Verzeichnis "T:" (befindet sich im RAM) abgelegt. Der Dateiname setzt sich dabei aus dem alten Dateinamen und dem aktuellen Modus, getrennt durch einen Punkt, zusammen. Eventuell im Namen vorhandene Leerzeichen werden entfernt. Die Datei »Test File« mutiert also beispielsweise zu »T:TestFile.Best« beziehungsweise »T:TestFile.Good« oder »T:TestFile.VeryGood«. Ist dies geschehen, verkündet ein weiterer Requester das Ergebnis der Aktion und fragt Sie, ob das Original mit der kürzesten Variante überschrieben werden soll oder nicht. Zum Schluß werden noch die drei temporären Dateien aus "T:" gelöscht.

Wie das Script intern arbeitet, sollte aus den sehr ausführlichen Kommentaren ersichtlich sein – deshalb hierzu nur folgendes: Ab sofort unterscheidet man automatisch zwischen den drei Packarten, je nach Länge der zu packenden Datei.

Zur besseren Eingabe haben wir das Script mit Zeilennummern versehen, die beim Eingeben natürlich nicht beachtet werden sollten.

Wie man sieht, ist ARexx als Steuer-Software wirklich zu gebrauchen. Vielleicht fallen Ihnen ja auch noch Anwendungsgebiete ein.



Christoph Teuber (jb)

```
************************************************************
*** Programm: PPTry            / Sprache: ARexx         ***
*** Autor: Christoph Teuber / (c) AMIGA DOS 1991/92 ***
************************************************************

/* PPTry 1.0 */

options results          /* notwendig, um die Ergebnisse
                            vom PowerPacker zu empfangen */

address PowerPacker  /* ab jetzt alle ARexx unbekannten
                        Befehle an den ARexx-Port mit
                        dem Namen Powerpacker          */


  file = "req"
  name = "dummy"
  BestLen = Try("Best")            /*                    */
  VeryGoodLen = Try("VeryGood") /* jed. Modus einmal */
  GoodLen = Try("Good")            /*                    */

  winner = max(BestLen, VeryGoodLen, GoodLen)  /* Die h
oechste Zahl gesparter
                                                   Bytes
 ermitteln */

  if winner = BestLen then         /* Und mit den Ergebnis
 sen der */
    Bestmode = Best                /* einzelnen Modi verge
 ichen */
  else
    if winner = VeryGoodLen then
      Bestmode = VeryGood
    else
      if winner = GoodLen then
        Bestmode = Good

  EZRequest Bestmode" war am effizientesten. Speichern
?"  /* Requester */
address command                /* ab jetzt unbekanntes an di
e Shell */

  if rc =5 then
      copy "t:"name"."Bestmode '"'file'"'  /* Wenn ok i
m Requester, kopieren */

  delete ">NIL: t:"name".best"                    /* temp
oraere Dateien loeschen */
  delete ">NIL: t:"name".good"
  delete ">NIL: t:"name".verygood"

exit

Try : procedure expose file name   /* Diese Procedur ue
bernimmt das Packen
                                      Die Bezeicher fil
e und name sind global,
                                      d.h. aenderungen
sind auch im
                                      Hauptprogramm sic
htbar */

  parse arg mode                      /* Argument (Modus)
uebernehmen */
  interpret "eff"mode                 /* Befehlszeile zusa
mmensetzen und
                                      als solche behand
eln, sprich
                                      interpretieren */

  if file = "req" then                /* Wenn noch keine D
atei gewaehlt */
    load                              /* Laden mit Filereq
uester */
  else
    load file                         /* Ansonsten alte Da
tei nehmen */

  won = SUBWORD(result,2,1)           /* Zahl der gewonnen
e Bytes aus dem
                                      Ergebnis von "loa
d" ermitteln */

  if file = "req" then do             /* Wenn aktuelle dat
ei aus Filerequester */
    GetFileName                       /* Name der Datei ho
len */
    name = COMPRESS(result)           /* Von Leerzeichen b
efreien und an name */
    GetFullName                       /* Kompletten Pfad h
olen */
    file = result                     /* Und an file */
  end
  mode = "t:"name"."mode              /* temporaeren Datei
namen zusammensetzen */
  save mode
  return won                          /* Zahl der gewonnen
en Bytes uebergeben */
```

Assembler

Zini – Super-Mauspfeil gefällig?

Wem der herkömmliche Mauspfeil zu langweilig ist, der sollte sich das Programm »Zini« einmal genauer ansehen.

Bei »Zini« handelt es sich um ein in Assembler geschriebenes Scherzprogramm, das eigentlich keinen Zweck erfüllt. Trotzdem dürfte es wohl für jeden wissensbegierigen Amiganer interessant sein, denn von (fast) jedem Programm kann einiges gelernt werden.

Zum Programm

Der Quelltext zu »Zini« läßt sich in einzelne Teile spalten und mit Überschriften versehen. Die folgende Gliederung wurde vorgenommen, damit das Nachvollziehen der einzelnen Programmschritte nicht zur Safari wird. Trotzdem ist es natürlich notwendig, den Quelltext genau zu studieren, wenn man das Programm verstehen will.

Gliederung des Quelltextes:
1. Quelltextkopf
2. Library-Offsets definieren
3. Konstanten definieren
4. »startup-sequence«, die das Starten des Programms von der Workbench und vom CLI aus möglich macht
5. Initialisierung (Libraries öffnen usw.)
6. Warteschleife
7. Programmende
8. Unterprogramme
9. Rasterstrahl-Interrupt
10. Daten (Strings usw.)

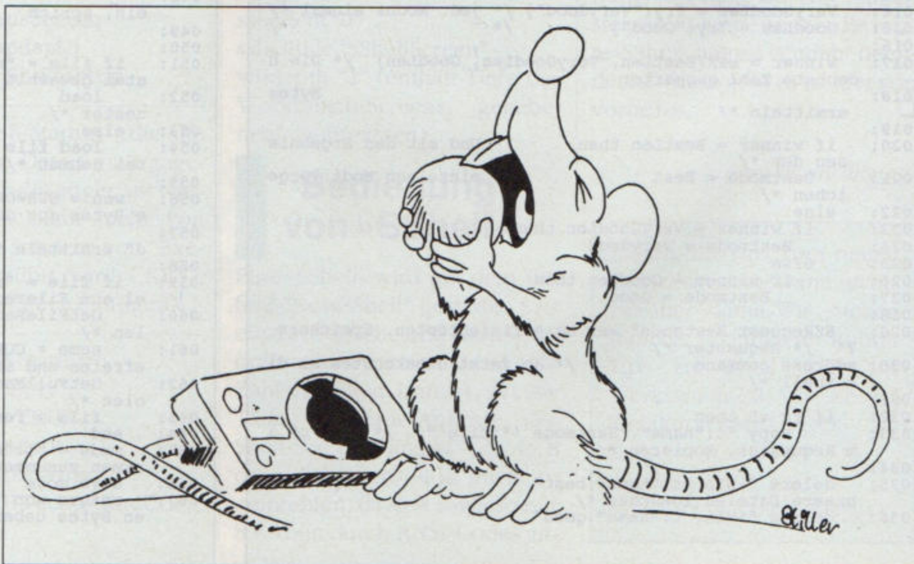
Aus dieser Gliederung des Quelltextes ergibt sich der folgende Programmablaufplan:

– Test, ob das Programm von der Workbench oder vom CLI aus gestartet wurde, und entsprechend reagieren
– Intuition-Library öffnen
– Graphics-Library öffnen
– Window öffnen
– Koordinatentabelle, in der die Positionen des Mauspfeils gespeichert werden, erstellen
– Speicher für Grafikdaten allokieren (reservieren)
– Mauspfeildaten ermitteln
– Farben der Sprites berechnen
– Simple-Sprite-Strukturen aufbauen
– Sprites allokieren (reservieren)
– Rasterstrahl-Interrupt, in dem die Koordinatentabelle aktualisiert wird, einschalten
– Warteschleife, die entweder durch Aktivieren des Close-(Schließ-)Gadgets (-> Ende) oder eines anderen Windows (-> Restart = Neustart) beendet wird
– Zini-Interrupt ausschalten
– Sprites freigeben
– Window schließen
– Speicher freigeben
– Libraries schließen
– Programmende -> Rücksprung auf AmigaDOS- oder Workbench-Ebene

Weiterhin muß gesagt werden, daß beim Auftreten eines Fehlers, zum Beispiel beim Allokieren (Reservieren) des Speichers, das Programm sofort beendet wird. Um ein Programm verstehen und nachvollziehen zu können, ist es unerläßlich, die benutzten Funktionen des Betriebssystems zu kennen. Aus diesem Grund folgt eine Liste von Funktionen, die das Sprite- und Interrupt-Handling betreffen und von dem Programm »Zini« verwendet werden. Auf die Darstellung der restlichen Funktionen wurde verzichtet, da dies den Umfang dieses Artikels sprengen würde.

Die Library-Funktionen

»Zini« gebraucht aus dem breiten Spektrum der Interrupt-Möglichkeiten des Amiga nur den Rasterstrahl-Interrupt. Dieser Interrupt, auch "Vertikal-Blank-Interrupt" genannt, wird immer dann ausgelöst, wenn sich der Rasterstrahl in der vertikalen Austastlücke befindet. Da der Aufbau des Bildes beim Amiga (PAL-Version) mit einer Frequenz von 50 Hertz erfolgt, wird der Rasterstrahl-Interrupt 50mal in der Sekunde aufgerufen. Durch diese Gleichmäßigkeit



kann dieser Interrupt besonders gut für eigene Routinen genutzt werden, bei denen es auf die zeitliche Abstimmung ankommt. Wie kann nun aber dieser Interrupt für eigene Zwecke genutzt werden? Die Exec-Library stellt uns dafür zwei Funktionen zur Verfügung:

AddIntServer(IntNummer,Interrupt)
-168 D0 A1
LONG IntNummer;
struct Node *Interrupt;

Die Funktion ergänzt einen weiteren Interrupt in der Interrupt-Server-Liste. "IntNummer" stellt den Interrupt-Typ dar, der in Register "D0" als Langwort übergeben werden muß. In unserem Fall wird "IntNummer" mit "INTB_VERTB" beziehungsweise mit "D0=5" (für Assembler-Programmierer) belegt. Das zweite Argument stellt einen Zeiger auf eine Interrupt-Struktur dar, die unten erläutert wird.

RemIntServer(IntNummer,Interrupt)
-174 D0 A1
LONG IntNummer;
struct Node *Interrupt;

Diese Funktion entfernt den angegebenen Interrupt aus der Interrupt-Server-Liste. Die Argumente sind die gleichen wie bei "AddIntServer", wobei die Interrupt-Struktur folgenden Aufbau aufweist:

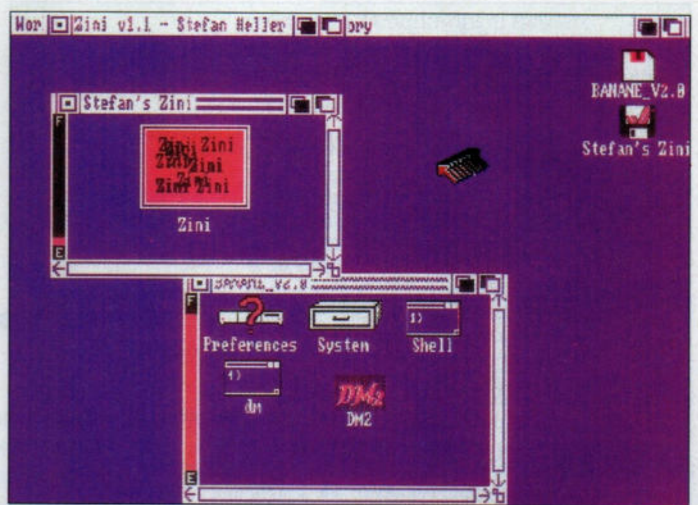
1struct Interrupt{
2 struct Node is_Node;
3 APTR is_Data;
4 VOID (*is_Code)();
5};

In Assembler sieht die Sache so aus:

1dc.l 0 ; Vorgänger und
2dc.l 0 ; Nachfolger (uninteressant)
3dc.b 2 ;Typ=NT_INTERRUPT
4dc.b 0 ;Priorität
5dc.l name ;Zeiger auf Name
6dc.l 0 ;Zeiger auf Daten (unwichtig)
7dc.l code ;Zeiger auf Code
8name:
9dc.b "ZinisInterrupt",0
10even

Der Code beziehungsweise die Interrupt-Routine muß am Ende das Zero-Flag (Z-Flag) setzen. Dies geschieht in Assembler mit:
MOVEQ #0,d0 RTS

Wie für die Interrupts die Interrupt-Struktur benötigt wird, so muß für die Sprites die Simple-Sprite-Struktur bekannt sein:



»Zini«, der spaßige Mauspfeil

1struct SimpleSprite { 2 UWORD *sprdata; /* Zeiger auf Grafikdaten */
3 UWORD height;
4 UWORD x,y; /* Position */
5 UWORD nummer; /* Nummer des Sprites */

Die einzelnen Funktionen zum Verwalten von Sprites befinden sich in der Graphics-Library. Mit

SpriteNr=GetSprite(Sprite,GewünschtesSprite)
D0 -408 A0 D0
SHORT SpriteNr;
struct SimpleSprite *Sprite;
SHORT GewünschtesSprite;

können Sie sich ein Hardware-Sprite zuweisen lassen. Die Parameter sind ein Zeiger auf eine Simple-Sprite-Struktur und die Nummer des gewünschten Sprites. Falls Ihnen die Nummer egal ist, können Sie -1 übergeben. "SpriteNr" enthält nach dem Funktionsaufruf die Nummer des zugewiesenen Sprites oder -1, wenn das gewünschte Sprite nicht allokiert (zugewiesen) werden konnte. Mit der Funktion

MoveSprite(ViewPort,Sprite x,y)
-426 A0 A1 D0 D1
struct ViewPort *ViewPort;
struct SimpleSprite *Sprite;
SHORT x,y;

wird das angegebene Sprite relativ zum "ViewPort" positioniert. Falls das Sprite relativ zum View bewegt werden soll, muß "ViewPort" auf null gesetzt werden. Die Hardware-Sprites müssen letztlich mit

FreeSprite(SpriteNr)
-414 D0
SHORT Spritenr;

wieder freigegeben werden, wenn Sie sie nicht mehr benötigen.

»Zini« paßt seine Form dem ursprünglichen Mauspfeil an, wenn sich das aktive Window ändert. Dies geschieht jedoch nicht, wenn innerhalb eines aktiven Fensters das Aussehen des Mauszeigers verändert wird. Vielleicht motiviert Sie das zu ähnlich gut gelungenen Umsetzungen. ❑

Stefan Heller (vb)

```
;
; Zini
; Autor: Stefan Heller
; Sprache:  Assembler Devpac 2.0
; (c) AMIGA DOS 1992
;

 INCDIR "df0:Include/"
 INCLUDE "exec/exec_lib.i"
 INCLUDE "graphics/graphics_lib.i"
 INCLUDE "intuition/intuition_lib.i"
 INCLUDE "misc/easystart.i"

; --- HAUPTPROGRAMM ---

zini lea iname,a1 ;Intuition-Library öffnen
 clr.l d0
 CALLEXEC OpenLibrary
 move.l d0,_IntuitionBase
 beq ende1
 movea.l d0,a0
```

```
 move.l 56(a0),win_scr ;ActiveScreen

 lea gname,a1 ;Graphics-Library öffnen
 clr.l d0
 CALLEXEC OpenLibrary
 move.l d0,_GfxBase
 beq ende2

 lea nwindow,a0  ;Window öffnen
 CALLINT OpenWindow
 move.l d0,window
 beq ende3

 movea.l _IntuitionBase,a0 ;Koordinatentabelle
 installieren
 lea kox,a1
 lea koy,a2
 moveq #13,d0
 move 70(a0),d1 ;x-Koordinate berechnen
 lsr d1
 move 68(a0),d2 ;y-Koordinate berechnen
```

```
 lsr d2
.loop move d1,(a1)+
 move d2,(a2)+
 dbf d0,.loop

schleif move.l #600,d0  ;Speicher allokieren
 move.l #$10002,d1 ;MEMF_CHIP|MEMF_CLEAR
 CALLEXEC AllocMem
 move.l d0,memory
 beq ende4

 bsr mausdat
 bsr sprfarb ;Farben festlegen
 bsr sprbere ;SimpleSprite-Strukturen aufbauen

getspr lea sprites,a5 ;Sprites einschalten
 moveq #1,d5
 moveq #6,d6
.loop move d5,d0 ;Spritenummer
 movea.l a5,a0 ;Adresse SimpleSprite-Struktur
 CALLGRAF GetSprite
 cmpi #-1,d0
 beq.s ende5
 move d0,10(a5) ;Sprite-Nummer in SimpleSprite-Struktur schreiben
 cmp.b d0,d5 ;gewünschtes Sprite ?
 bne.s ende5 ;nein -> Ende
adda.l #12,a5 ;Zeiger auf nächste SimpleSprite-Struktur
 addq #1,d5 ;Spritenummer um 1 erhöhen
 dbf d6,.loop

 moveq #5,d0 ;Interrupt ein
 lea intstru,a1 ;d0=5 -> Raster-Interrupt
 CALLEXEC AddIntServer

warten bsr movespr  ;Warteschleife
 movea.l _IntuitionBase,a0
 move.l 52(a0),d0 ;ActiveWindow
 cmp.l awin,d0
 beq.s .next
 bsr altint
 bsr freispm
 bra schleif
.next movea.l window,a0
 movea.l $56(a0),a0 ;User-Port
 CALLEXEC GetMsg
 tst.l d0
 beq.s warten
 movea.l d0,a1
 CALLEXEC ReplyMsg

ende bsr altint
ende5 bsr freispm
ende4 movea.l window,a0 ;Window schließen
 CALLINT CloseWindow
ende3 movea.l _GfxBase,a1 ;Libs schließen
 CALLEXEC CloseLibrary
ende2 movea.l _IntuitionBase,a1
 CALLEXEC CloseLibrary
ende1 clr.l d0
 rts

; --- UNTERPROGRAMME ---

mausdat movea.l _IntuitionBase,a0  ;Daten Mauspfeil lesen
 move.l 52(a0),awin
 clr d0
 move.b 118(a0),d0 ;Hot-Spot Koordinaten lesen
 not.b d0
 addq.b #1,d0
 move d0,xoff
 move.b 119(a0),d0
 not.b d0
 addq.b #1,d0
 move d0,yoff
 rts

sprfarb movea.l _IntuitionBase,a4  ;Farben festlegen
 movea.l 56(a4),a3
 adda.l #$2c,a3  ;ViewPort-Struktur
 moveq #11,d4 ;12 Farben
 moveq #16,d5 ;Farbregister 16
.loop move d5,d0  ;Farbregisternummer
 movea.l 4(a3),a0 ;Color-Map
 CALLGRAF GetRGB4
 move d0,d1 ;Rotanteil
 lsr #8,d1
 andi.b #$f,d1
 subq.b #3,d1
 bcc.s .next1
 clr.b d1
.next1 move d0,d2 ;Grünanteil
 lsr #4,d2
 andi.b #$f,d2
 subq.b #3,d2
 bcc.s .next2
 clr.b d2
.next2 move d0,d3 ;Blauanteil
 andi.b #$f,d3
 subq.b #3,d3
 bcc.s .next3
 clr.b d3
.next3 move d5,d0 ;Farbregisternummer
 addq #4,d0
 movea.l a3,a0  ;ViewPort-Struktur
 CALLGRAF SetRGB4
 addq #1,d5
 dbf d4,.loop
 rts

sprbere movea.l _IntuitionBase,a4  ;SimpleSprite-Strukturen aufbauen
 movea.l $70(a4),a3
 movea.l memory,a0
 lea sprites,a1  ;Adresse der 1. SimpleSprite-Struktur
 clr d0
 move.b $74(a4),d0 ;Pointer-Höhe aus _IntuitionBase-Struktur
 addq #1,d0
 moveq #6,d1
.loop1 move.l a0,(a1)  ;Adresse  Spritedaten eintragen
 move d0,4(a1)  ;Höhe eintragen
 move #-1,10(a1)
 movea.l a3,a2
 move d0,d2
 mulu #4,d2
 subq #1,d2
.loop2 move.b (a2)+,(a0)+ ;Daten kopieren
 dbf d2,.loop2
 clr (a0)+  ;zwei Nullworte anhängen (wichtig !)
 clr (a0)+
 adda.l #12,a1  ;Zeiger auf nächste SimpleSprite-Struktur
 dbf d1,.loop1
 rts

movespr lea kox,a3  ;Sprites darstellen
 lea koy,a4
 lea sprites,a5  ;Adresse der 1. SimpleSprite-Struktur
 moveq #2,d4
 moveq #6,d5 ;7 Sprites
.loop movea.l #0,a0 ;a0=0 -> aktueller View
 movea.l a5,a1 ;Zeiger auf SimpleSprite-Struktur
 move 10(a5),d2  ;Spritenummer
 move 0(a4,d4),d1 ;y-Koordinate berechnen
 sub yoff,d1
 sub d2,d1
 move 0(a3,d4),d0 ;x-Koordinate berechnen
 sub xoff,d0
 lsl d2
 add d2,d0
 CALLGRAF MoveSprite
 adda.l #12,a5  ;nächste SimpleSprite-Struktur
 addq.l #4,d4  ;Koordinaten-Tabellenindex erhöhen
 dbf d5,.loop
 rts

altint moveq #5,d0  ;Interrupt aus
 lea intstru,a1
 CALLEXEC RemIntServer
 rts

freispm lea sprites,a4  ;Sprites freigeben
 moveq #6,d4
.loop move 10(a4),d0 ;Spritenummer
 cmpi #-1,d0
 beq.s .next
 CALLGRAF FreeSprite
.next adda.l #12,a4
 dbf d4,.loop
 movea.l memory,a1 ;Speicher freigeben
 move.l #600,d0
 CALLEXEC FreeMem
 rts

; --- INTERRUPT ---

meinint lea koye,a0 ;Koordinaten verschieben
 moveq #26,d0 ;27 mal
.loop move (a0)+,d1
 move d1,(a0)
 suba.l #4,a0
 dbf d0,.loop
 movea.l _IntuitionBase,a0 ;aktuelle Koordinate eintragen (an 1. Stelle)
 move 70(a0),d0 ;x-Koordinate berechnen
 lsr d0
 move d0,kox
 move 68(a0),d0 ;y-Koordinate
 lsr d0
 move d0,koy
 moveq #0,d0
 rts

; --- DATEN ---

iname INTNAME
gname GRAFNAME
 even
_IntuitionBase dc.l 0
_GfxBase dc.l 0
memory dc.l 0
awin dc.l 0
window dc.l 0

intstru dc.l 0,0 ;Interrupt-Struktur
 dc.b 2,0
 dc.l intname,0,meinint
intname dc.b "Zini",0,0

nwindow dc.w 30,0,289,10 ;Window-Struktur
 dc.b -1,-1
 dc.l 512,$100e,0,0,winname
win_scr dc.w 0,0,0,0,0,0,0,0,15
winname dc.b "Zini v1.1 - Stefan Heller",0

sprites ds.b 7*12
kox ds.w 14
koy ds.w 12
koye ds.w 2
xoff dc.w 0
yoff dc.w 0 ;... Ende
```

AmigaBASIC

Kniffelspaß mit Bongo-Bongo

Denk- und Knobelspielchen üben von jeher einen ganz besonderen Reiz aus.

Dieses in AmigaBASIC geschriebene Spiel ist der Kategorie Denkspiele zuzuordnen. Es stellt gleichzeitig ein sehr gutes Beispiel dar, wie man in dieser Programmiersprache strukturiert Programme entwickelt. Doch nun zum Spiel selbst: Zur Steuerung wird der Joystick (Port b) benötigt und, falls auf dem Amiga eine Tastatur vorhanden sein sollte, auch diese. Nach dem Start erfolgt eine kleine Animation, die mit der Leertaste beendet wird. Nach einer etwas längeren Aufbauphase erscheint das Spielfeld. Mit dem "Freudenknüppel" haben Sie nun die Möglichkeit, zwischen vier Schwierigkeitsgraden und insgesamt zehn Level zu wählen. Danach wird das Menü ausgeschaltet, und das Spiel kann beginnen.

Sinn von »Bongo-Bongo« ist es, die angezeigte Symbolformation auf dem Spielfeld nachzubilden. Bei jeder Bewegung des Cursors drehen sich die äußeren Symbole innerhalb der Position des Cursors im Uhrzeigersinn um eine Position weiter. Dabei gibt es natürlich verschiedene Schwierigkeitsgrade:

1. Die Formation muß innerhalb des rot eingerahmten Quadrates erzeugt werden. Mit dem Feuerknopf sind weitere Drehbewegungen möglich.
2. Wie (1.). Feuerknopf außer Betrieb.
3. Formation muß innerhalb des Cursors vorliegen. Feuerknopf in Betrieb.
4. Wie (3.). Feuerknopf außer Betrieb.

Mit "Q" können Sie jederzeit das Programm beenden. Viel Spaß beim Spielen. ❑

Wolfgang Serafin (vb)

Info für Neueinsteiger

– Laden Sie Ihre Extras-Diskette (Workbench 1.3).
– Klicken Sie mit einem Doppelklick AmigaBASIC an.
– Ziehen Sie mit der linken Maustaste das List-Fenster nach links, und vergrößern Sie das Fenster, indem Sie ebenfalls mit der linken Maustaste auf die rechte untere Ecke klicken.
– Posionieren Sie den Cursor links oben im List-Fenster.
– Danach können Sie das Listing abtippen, jedoch **ohne** die Zeilennummern mit einzugeben.
– Zum Speichern gehen Sie in die Statuszeile und betätigen die linke Maustaste. Es wird ein Disk-Menü geöffnet, in dem Sie die Menüpunkte "Project", "Edit", "Run" und "Window" vorfinden. Fahren Sie nun mit der Maus im Menü "Project" auf den Menüpunkt "Save As". Daraufhin öffnet sich ein Requester, und Sie können den Namen des Programms, mit entsprechender Pfadangabe, eingeben.
Zum Beispiel: df0:bongo-bongo.bas
Wenn Sie mit der Maus auf "OK" klicken, wird das Programm gespeichert.
– Wollen Sie das Programm starten, gehen Sie mit der Maus im Menü "Run" auf den Menüpunkt "Start".

```
REM
REM Bongo-Bongo
REM Autor: Wolfgang Serafin
REM Sprache: AmigaBASIC
REM (c) AMIGA DOS 1992
REM

SCREEN 1,320,256,3,1
WINDOW 2,,(0,0)-(311,235),16,1
DEFINT a-z
DIM SHARED zeichen(29),bild1(116),bild2(116),bild3(116)
DIM SHARED bild4(116),bild5(116),level$(10),tit(128)
daten
bilder
farben
titel
schwarz
maske
farben
xc=25:yc=50:level=1:s=1
CALL menue (level,s)

start:
IF UCASE$(INKEY$)="Q" THEN:WINDOW CLOSE 2:SCREEN CLOSE
 1:SYSTEM
xj=STICK(2)
yj=STICK(3)
IF s=1 OR s=3 THEN zj=STRIG(3)
IF xj=0 AND yj=0 AND zj=0 THEN start
wt=0
IF xj=1 AND xc<105 THEN sline xc,yc:xc=xc+20:sl xc,yc,s
 :wt=1
IF xj=-1 AND xc>25 THEN sline xc,yc:xc=xc-20:sl xc,yc,s
 :wt=1
IF yj=1 AND yc<125 THEN sline xc,yc:yc=yc+20:sl xc,yc,s
 :wt=1
IF yj=-1 AND yc>50 THEN sline xc,yc:yc=yc-20:sl xc,yc,s
 :wt=1
fehler=1
IF s=1 THEN
  fehler=0
  CALL drehen (xc,yc,s,level,fehler)
END IF
IF s=2 AND wt=1 THEN
  fehler=0
  CALL drehen (xc,yc,s,level,fehler)
END IF
IF s=3 THEN
  fehler=0
  CALL drehen (xc,yc,s,level,fehler)
END IF
IF s=4 AND wt=1 THEN
  fehler=0
  CALL drehen (xc,yc,s,level,fehler)
END IF
IF fehler=0 THEN:CALL newplay(level,s)
GOTO start

SUB drehen (x,y,s,l,f) STATIC
le$="":dr$="":aw$="":we$="":RESTORE
xa=x+10:xe=x+50:ya=y+1:ye=y+41
FOR i= xa TO xe STEP 20
  FOR k= ya TO ye STEP 20
    pu=POINT(i,k)
    CALL bestimmen (pu,aw$)
    le$=le$+aw$
  NEXT k
NEXT i
FOR i=1 TO 9
READ a
dr$=dr$+MID$(le$,a,1)
NEXT i
setz=1
FOR i=x+1 TO x+41 STEP 20
  se=VAL(MID$(dr$,setz,1))
  setbild i,y+1,se
  setz=setz+1
NEXT i
FOR i=y+21 TO y+41 STEP 20
  se=VAL(MID$(dr$,setz,1))
  setbild x+41,i,se
  setz=setz+1
NEXT i
FOR i=x+21 TO x+1 STEP -20
  se=VAL(MID$(dr$,setz,1))
  setbild i,y+41,se
  setz=setz+1
NEXT i
se=VAL(MID$(dr$,setz,1))
setbild x+1,y+21,se
IF s<3 THEN
xa=75:xe=115:ya=91:ye=131
ELSE
xa=x+10:xe=x+50:ya=y+1:ye=y+41
END IF
FOR i= xa TO xe STEP 20
  FOR k= ya TO ye STEP 20
    pu=POINT(i,k)
    CALL bestimmen (pu,a$)
    we$=we$+a$
  NEXT k
NEXT i
```

```
FOR i=1 TO 9
  IF MID$(level$(l),i,1) <> MID$(we$,i,1) THEN f=1
NEXT i
END SUB
DATA 2,1,4,7,8,9,6,3,5
SUB setbild (xp,yp,b) STATIC
IF b=1 THEN PUT(xp,yp),bild1,PSET
IF b=2 THEN PUT(xp,yp),bild2,PSET
IF b=3 THEN PUT(xp,yp),bild3,PSET
IF b=4 THEN PUT(xp,yp),bild4,PSET
IF b=5 THEN PUT(xp,yp),bild5,PSET
END SUB
SUB bestimmen (p,aw$) STATIC
IF p=5 THEN aw$="1"
IF p=6 THEN aw$="2"
IF p=7 THEN aw$="3"
IF p=4 THEN aw$="4"
IF p=0 THEN aw$="5"
END SUB
SUB sline (x,y) STATIC
LINE(x,y)-(x+60,y+60),4,b
END SUB
SUB sl (x,y,s) STATIC
IF s<3 THEN c=7 ELSE c=4
LINE(65,90)-(125,150),c,b
LINE (x+15,y)-(x,y),6
LINE -(x,y+15),6
LINE (x+45,y)-(x+60,y),6
LINE -(x+60,y+15),6
LINE (x,y+45)-(x,y+60),6
LINE -(x+15,y+60),6
LINE (x+45,y+60)-(x+60,y+60),6
LINE -(x+60,y+45),6
END SUB
SUB setlevel (l) STATIC
z=1
FOR i=221 TO 261 STEP 20
  FOR k=51 TO 91 STEP 20
    b=VAL(MID$(level$(l),z,1))
    setbild i,k,b
    z=z+1
  NEXT k
NEXT i
END SUB
SUB newplay (l,s) STATIC
l=l+1
IF l=11 THEN l=1:s=s+1
IF s=5 THEN:spielende
LOCATE 21,36:PRINT USING"#";s
LOCATE 23,35:PRINT USING"##";l
FOR m=160 TO 420 STEP 20
  SOUND m,3
NEXT m
FOR m=400 TO 160 STEP -20
  SOUND m,3
NEXT m
setlevel l
END SUB
SUB menue (l,s) STATIC
y=158:se=0:e=0:setmenue y,l,s,e
st:
xj=STICK(2)
yj=STICK(3)
zj=STRIG(3)
IF yj=1 AND y<175 THEN y=y+16
IF yj=-1 AND y>159 THEN y=y-16
IF xj=0 AND yj=0 AND zj=0 THEN st
IF xj=1 AND y=158 AND s<4 THEN s=s+1
IF xj=-1 AND y=158 AND s>1 THEN s=s-1
IF xj=1 AND y=174 AND l<10 THEN l=l+1
IF xj=-1 AND y=174 AND l>1 THEN l=l-1
setmenue y,l,s,e
setlevel l
IF zj<>-1 OR y<>190 THEN st
sl 25,50,s
e=1
setmenue y,l,s,e
END SUB
SUB setmenue (y,l,s,e) STATIC
FOR i=158 TO 190 STEP 16
  IF y=i THEN col=5 ELSE col=1
  LINE(214,i)-(289,i+11),col,b
NEXT i
LOCATE 21,36:PRINT USING"#";s
LOCATE 23,35:PRINT USING"##";l
IF e=1 THEN :p$="OFF":LOCATE 19,34:PRINT USING"&";p$
END SUB
SUB daten STATIC
level$(1)="111222333"
level$(2)="431431431"
level$(3)="222242222"
level$(4)="155515551"
level$(5)="444424444"
level$(6)="323222323"
level$(7)="111111555"
level$(8)="443443333"
level$(9)="111111111"
level$(10)="555555555"
END SUB
SUB maske STATIC
LOCATE 1,1:COLOR 7:PRINT "BONGO-BONGO"
LINE(41,4)-(46,4),7
FOR i=0 TO 88
  GET(i,0)-(i,7),zeichen
  FOR k=0 TO 2
    PUT(26+z+k,15),zeichen,PSET
  NEXT k
  z=z+3
NEXT i
FOR i=20 TO 0 STEP -2
  SCROLL (0,0)-(320,i),0,-1
NEXT i
DIM feld(7)
feld(0)=&H82A0
feld(1)=&H4141
feld(2)=&HA082
feld(3)=&H580D
feld(4)=&HA082
feld(5)=&H4141
feld(6)=&H82A0
feld(7)=&H8D58
PATTERN &HFFFF,feld
PAINT(0,0),1,7
ERASE feld
DIM feld(7)
FOR i=0 TO 7
  feld(i)=&HFFFF
NEXT i
PATTERN &HFFFF,feld
LINE(24,49)-(166,191),0,bf
LINE(219,49)-(281,111),0,bf
FOR i=25 TO 165 STEP 20
  LINE(i,50)-(i,190),4
NEXT i
FOR i=50 TO 190 STEP 20
  LINE(25,i)-(165,i),4
NEXT i
schalter 167,23,192,48,15
FOR i=220 TO 280 STEP 20
  LINE(i,50)-(i,110),4
NEXT i
FOR i=50 TO 110 STEP 20
  LINE(220,i)-(280,i),4
NEXT i
schalter 282,218,112,48,15
b=1
FOR i=26 TO 146 STEP 20
  FOR k=51 TO 171 STEP 20
    setbild i,k,b
    b=b+1
    IF b=6 THEN b=1
  NEXT k
NEXT i
LINE(203,135)-(300,208),1,bf
LINE(203,215)-(300,240),1,bf
LINE(19,215)-(172,240),1,bf
ERASE feld
DIM feld(3)
feld(0)=&HCCCC
feld(1)=&HCCCC
feld(2)=&H3333
feld(3)=&H3333
PATTERN &HFFFF,feld
LINE(208,140)-(295,203),2,bf
LINE(208,220)-(295,235),2,bf
LINE(24,220)-(167,235),2,bf
ERASE feld
DIM feld(3)
FOR i=0 TO 3
  feld(i)=&HFFFF
NEXT i
PATTERN &HFFFF,feld
COLOR 6
LOCATE 29,28:PRINT" Q"
LOCATE 29,30:COLOR 5:PRINT " u i t "
LOCATE 19,28:PRINT "Menue  ON"
LOCATE 21,28:PRINT "Grad    1"
LOCATE 23,28:PRINT "Level   1"
LOCATE 25,28:PRINT " P L A Y "
FOR i=142 TO 190 STEP 16
  LINE(214,i)-(289,i+11),1,b
NEXT i
LINE(214,222)-(289,233),1,b
LINE(30,222)-(161,233),1,b
LOCATE 29,5:PRINT " (c)  AMIGA DOS "
setlevel 1
END SUB
SUB schalter (r,l,u,o,fak)STATIC
COLOR 4
AREA(r,o):AREA(r+fak,o-fak):AREA(r+fak,u+fak):AREA(r,u)
:AREA(r,o):AREAFILL
COLOR 1
AREA(l,u):AREA(r,u):AREA(r+fak,u+fak):AREA(l-fak,u+fak)
:AREA(l,u):AREAFILL
COLOR 3
AREA(l-fak,o-fak):AREA(l,o):AREA(l,u):AREA(l-fak,u+fak)
AREA(l-fak,o+fak):AREAFILL
COLOR 2
AREA(l-fak,o-fak):AREA(r+fak,o-fak):AREA(r,o):AREA(l,o)
AREA(l-fak,o-fak):AREAFILL
END SUB
SUB spielende STATIC
LOCATE 29,5:PRINT "G A M E  O V E R"
WHILE INKEY$ =""
RANDOMIZE TIMER
```

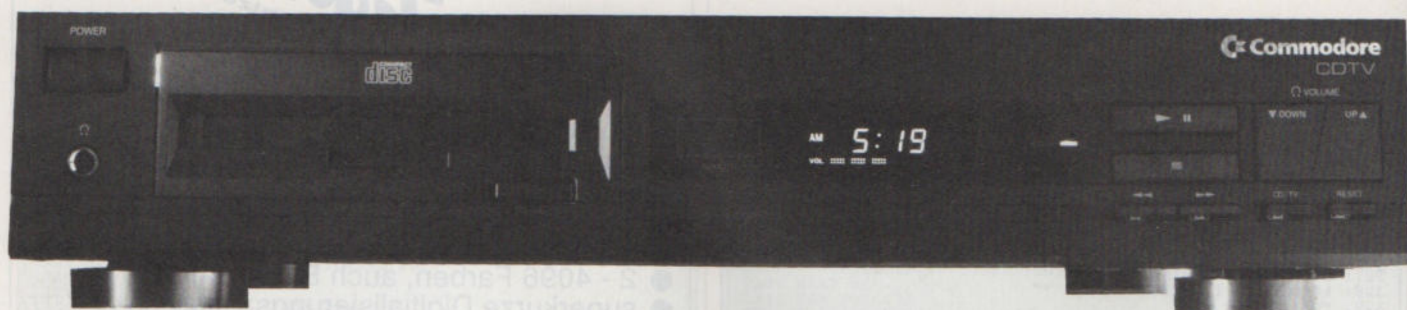
```
x=26+((INT(RND*7))*20)
y=51+((INT(RND*7))*20)
b=1+INT(RND*5)
setbild x,y,b
WEND
WINDOW CLOSE 2:SCREEN CLOSE 1:SYSTEM
END SUB

SUB titel STATIC
COLOR 6
LOCATE 8,9:PRINT "BONGO-BONGO by W.Serafin"
LOCATE 10,11:PRINT "(c) 1991  AMIGA DOS
COLOR 5
LOCATE 9,18:PRINT "* * *"
LOCATE 11,17:PRINT "* * * *"
FOR i=1 TO 7
  CIRCLE(156,69),140+i,i,,,.2+(i/50)
NEXT i
f=0
FOR i=15 TO 123
  GET(0,i)-(320,i+1),tit
  PUT(0,235-f),tit,PSET
  f=f+1
NEXT i
PUT(40,60),bild4,PSET
PUT(260,60),bild5,PSET
WHILE INKEY$ =""
  GET(60,50)-(260,51),tit
  SCROLL(60,50)-(260,86),0,-1
  PUT(60,86),tit,PSET
WEND
FOR i=1 TO 255 STEP 2
  LINE(0,i)-(320,i),0
  LINE(0,255-i)-(320,255-i),0
NEXT i
END SUB

SUB bilder STATIC
f1=5:f2=6:f3=7
CIRCLE(9,9),9,f1,-3.5,-2.8,1
PAINT(7,5),f1,f1
CIRCLE(10,8),6,f3,2,2.5,1
CIRCLE(10,9),6,f3,2,2.5,1
CIRCLE(10,5),4,f2,5.6,.5,1
CIRCLE(8,13),6,f3,2,2.8,1.2
CIRCLE(8,14),6,f3,2,2.8,1.2
GET(0,0)-(18,18),bild1
CLS
CIRCLE(9,9),9,f2,,,1
PAINT(9,9),f2,f2
CIRCLE(9,10),5,f3,3.14,0,1
CIRCLE(5,5),1,f3,,,1:PSET(5,5),f3
CIRCLE(13,5),1,f3,,,1:PSET(13,5),f3
COLOR f1
AREA(9,7):AREA(7,11):AREA(11,11):AREA(9,7):AREAFILL
GET(0,0)-(18,18),bild2
CLS
COLOR f3
AREA(5,0):AREA(0,5):AREA(8,18):AREA(10,18):AREA(18,5)
AREA(13,0):AREA(5,0):AREAFILL
LINE(4,4)-(6,6),2,bf
LINE(12,4)-(14,6),2,bf
CIRCLE(9,10),5,f3,0,3.14,1
CIRCLE(9,13),3,1,3.14,0,1
COLOR 1
AREA(9,6):AREA(7,11):AREA(11,11):AREA(9,6):AREAFILL
GET(0,0)-(18,18),bild3
CLS
COLOR f3
AREA(9,9):AREA(3,18):AREA(15,18):AREA(9,9):AREAFILL
CIRCLE(9,9),6,f2,,,1
PAINT(9,9),f1,f2
LINE(5,0)-(13,5),4,bf
PSET(7,8),0:PSET(11,8),0
CIRCLE(9,14),3,0,3.14,0,1
CIRCLE(9,11),2,1,3.14,0,1
LINE(9,9)-(9,10),f3
GET(0,0)-(18,18),bild4
CLS
COLOR f1
AREA(9,9):AREA(3,18):AREA(15,18):AREA(9,9):AREAFILL
CIRCLE(9,9),4,4,,,1
PAINT(9,9),4,4
LINE(6,1)-(12,5),f3,bf
LINE(5,5)-(13,5),f3
PSET(7,8),0:PSET(11,8),0
CIRCLE(9,14),3,0,3.14,0,1
CIRCLE(9,11),2,1,3.14,0,1
LINE(9,9)-(9,10),f3
PSET(4,8),f2:PSET(14,8),f2
GET(0,0)-(18,18),bild5
CLS
END SUB

SUB schwarz STATIC
FOR i=0 TO 7
  PALETTE i,0,0,.4
NEXT i
END SUB

SUB farben STATIC
PALETTE 0,0,0,.4
PALETTE 1,0,0,1     'blau0
PALETTE 2,0,0,.67    'blau1
PALETTE 3,0,.27,.8     'blau2
PALETTE 4,0,.4,1      'blau3
PALETTE 5,.6,.6,.6  'grau
PALETTE 6,1,1,0     'gelb
PALETTE 7,1,0,0     'rot
END SUB
```



CDTV-Player intern

Programmierung in AmigaBASIC

Insider wissen, daß sich im CDTV-Player ein kompletter Amiga verbirgt, den man genau wie jeden Amiga auch programmieren kann.

Aber ein paar Sachen sind doch anders, und das soll hier einmal praktisch erläutert werden.
Wenn man lediglich den CDTV-Player besitzt, kann man allerdings mangels Tastatur und beschreibbarem Speichermedium schlecht programmieren. Um das folgende nachzuvollziehen, brauchen Sie als Mindestausstattung einen CDTV-Player, einen Amiga mit 1.3-Betriebssystem (nur dort ist das AmigaBASIC dabei) und ein externes Diskettenlaufwerk, das an den CDTV-Player angeschlossen wird, um Ihre Programme aufzunehmen. Alternativ geht natürlich auch ein CDTV-Player mit angeschlossener Tastatur und dem Diskettenlaufwerk. Dann ist er ja praktisch ein kompletter Amiga.
Das Diskettenlaufwerk eröffnet mit einem Schlag schon einen riesigen Teil der vorhandenen Amiga-Software: Da die standardmäßige Fernbedienung wie die normale Amiga-Maus arbeiten kann (nur etwas ungenauer), läßt sich vollständig mausbediente Software klaglos benutzen.
Der Teufel liegt allerdings oft im Detail: Ab und zu muß man doch einmal Tastatureingaben machen, zum Beispiel um seinen Namen in die Highscore-Liste einzutragen. Und da merkt man, daß die Fernbedienung recht wenig Tasten hat.
Was steht uns nun auf der Fernbedienung an Tasten zur Verfügung, auf die man in der Programmierung zurückgreifen kann? Nun, neben den beiden Maustasten (A für linke und B für rechte) gibt es:
– eine Enter-Taste [Return]
– zehn Ziffertasten [0] bis [9]
– eine Escape-Taste [Esc]
Letztere liefert auch den ASCII-Code 27. Die Programmierrichtlinien besagen aber, daß man sie allgemein eher als Hilfe-Taste verwenden soll. Die Beschriftung der Taste wird eventuell später einmal auf ein Fragezeichen geändert. Aber auf jeden Fall wissen wir, daß wir sie mit Standardmitteln mit dem genannten Code abfragen können.

Ferngesteuerter CDTV-Player

Wer also keine Buchstabentasten braucht, kann im reinen Mausbetrieb schon fast ein normales Amiga-Feeling bekommen. Die Workbench hilft beim Kopieren von Dateien und Disketten oder auch beim Löschen.
Und mit Utilities wie »DirMaster« oder »DirOpus« oder »SID« kann man dies noch wesentlich gründlicher und zum Beispiel auch Bild- oder Animationsdateien ansehen. Eine eingelegte Programm- beziehungsweise Daten-CD ist immer als »CD0:« wie eine normale Diskette oder Festplatte ansprechbar, nur nicht beschreibbar.
Was ist neben der Fernbedienung noch an Besonderheiten zu beachten? Nun, vor allem ist das CDTV-Betriebssystem-ROM doppelt so groß wie ein herkömmliches Kickstart-1.3-ROM. Die eine Hälfte enthält praktisch den normalen 1.3-Kickstart, die andere viele Sachen zusätzlich, die im normalen 1.3-System entweder von der Workbench-Diskette nachgeladen werden müssen oder gar nicht vorhanden sind.
Ich will hier nur auf einen Systemteil eingehen, der es uns erlaubt, den Audio-CD-Teil anzusteuern. Im Multitasking können wir dann im Endeffekt zum Beispiel eine fetzige Amiga-Grafikdemo laufen lassen, während ein anderes Programm steuert, was gerade über die Stereoanlage dazu von einer Audio-CD gespielt werden soll.
Andere Libraries im CDTV-ROM dienen zum Beispiel zur Ansteuerung der Speicherkartenschnittstelle, wo scheckkartengroße, nichtflüchtige RAM-Karten eingesteckt werden. Sie können mit standardmäßigen Amiga-Device-Befehlen angesprochen werden.
Im Beispielprogramm, das im Listing abgedruckt ist, wird also die Anwendung des »cdtv.device« gezeigt, mit dem man vom Amiga-Teil aus den CD-Audio-Teil steuern kann. Wie immer, wenn es um die Programmierung von Amiga-Devices geht, habe ich als Ausgangspunkt das Programm »ScreenPrint« benutzt, das in den BASIC-Demos der Diskette Extras 1.3 zu finden ist. Hier finden sich sauber strukturiert die Programmteile zum Initialisieren der benötigten Datenstrukturen, zum Öffnen des Devices selbst, zum Absetzen eines Befehls und schließlich zum sauberen Aufräumen aller reservierten Betriebsmittel bei Programmschluß.
In diesen grundlegenden Device-Routinen mußte ich vor allem den Namen des Devices ändern. Darüber hinaus mußten aber auch zwei unabhängige Befehlskanäle zum »cdtv.device« eröffnet werden, was durch Einfügung entsprechender Laufwerksanweisungen und Array-Variablen geschah. Das ist notwendig, weil ein Kanal permanent belegt ist, solange eine Audio-CD auf Wiedergabe läuft. Will man währenddessen den aktuellen Status abfragen (Ist das aktuelle Stück, die "Spur", schon zu Ende?) oder auch die Lautstärke ändern, so braucht man dafür einen zweiten Kanal.
Bei Programmstart wird zunächst mit dem Kommando »CD.ISROM« abgefragt, ob die eingelegte CD eventuell ein CD-ROM ist. Bei einer Audio-CD darf das natürlich nicht der Fall sein. Andernfalls wird mit »CD.TOCMSF« (TOC = Table Of Contents = Inhaltsverzeichnis) die Anzahl und Dauer der einzelnen Spuren (Stücke) ermittelt.

Man erhält die Anfangszeitdaten in Minuten und Sekunden vom Anfang der CD ab gerechnet. Ich habe sie lieber auf die Dauer der einzelnen Stücke umgerechnet. Die Spuren zählen ab Nummer 1, nicht ab "0". Es gibt im Inhaltsverzeichnis aber auch einen "nullten" Eintrag, wo wir die Gesamtdauer der CD und die Anzahl der vorhandenen Spuren finden. Mit »CD.FADE« wird schließlich noch standardmäßig die volle Lautstärke eingestellt.

Bei der Marke "Eingabe:" tritt das Programm in den Teil ein, wo man die gewünschte Spur wählen oder auch das Programm verlassen kann. Die Bedienung erfolgt über Zahleneingaben für die Spurnummer und Quasi-Gadgets, also anklickbare Rechteckbereiche, mit denen man etwas auslösen kann. Statt "echter" Gadget-Programmierung wird hier mit vorhandenen BASIC-Befehlen nur ein Mausklick erkannt und dann anhand der Koordinaten festgestellt, ob das in einem unserer Quasi-Gadgets passierte. Folgende Eingabemöglichkeiten hat man:

- Zifferntasten: Eingabe der Spurnummer (Spur 0 heißt Programmende)
- Enter-Taste: Start des Abspielens der Spur, Anklicken des Wortes "Spur" hat die gleiche Wirkung
- Escape-Taste: Programmende
- Anklicken "Ende": Programmende
- Anklicken +: nächste Spur
- Anklicken -: vorherige Spur
- Anklicken +10: zehn Spuren weiter
- Anklicken -10: zehn Spuren zurück
- Anklicken "Leise": Leise stellen
- Anklicken "Endlos": Umschalten auf Modus "Einzeln"

Die Quasi-Gadgets "Leise" und "Endlos/Einzeln" sind Umschalter, ein nochmaliges Anklicken stellt wieder den anderen Zustand her. Das Leise- beziehungsweise Lautstellen geschieht mit »CD.FADE«, also nicht schlagartig, sondern mit sanftem Übergang. Im "Endlos-Betrieb" wird durchgespielt, am Ende der CD wird sofort wieder mit Spur 1 weitergemacht. Im "Einzeln-Betrieb" wird dagegen nach jedem Stück unterbrochen. Bei jeder Änderung der Spurnummer wird sofort die Dauer des gerade angezeigten Stücks ausgegeben. Unmögliche Spurnummern werden nicht angenommen.

Wenn man dann mit der Enter-Taste oder durch Anklicken von "Spur" das Abspielen gestartet hat, wechselt in der Anzeige das Quasi-Gadget "Ende" auf "Abbruch"; mit ihm kann jetzt das aktuelle Stück abgebrochen werden. Das Starten geschieht mit dem Befehl »CD.PLAYTRACK«. Unter der Anzeige der Gesamtdauer des Stücks erscheint eine zusätzliche Anzeige der davon bereits abgelaufenen Zeit. Dieser aktuelle Status wird mit »CD.SUBQMSF« ermittelt. Hier bekommt man außer der Zeit auch noch die Nummer der aktuellen Spur und eine Statusvariable. An ihnen kann man erkennen, ob schon ein neues Stück begonnen hat oder ob ein Problem vorliegt. Da erfahrungsgemäß viele CDs unterwegs kleine Probleme haben, die sich gar nicht fürs Ohr bemerkbar machen, wird erst abgebrochen, wenn dreimal hintereinander ein abweichender Status erkannt wird. Sie können das selbst an der Anzeige der abgelaufenen Zeit erkennen: Sie springt mitten im Stück manchmal auf "0" zurück; hier hat es ein kleines Problem mit der CD gegeben.

Fortsetzung auf Seite 107

Das Tier

Die Rückkehr des Bösen

NEU

Das Text/Grafik-Adventure der Horrorklasse – komplett in deutsch

Folgen Sie Graf Friedrich von Crasts Einladung auf sein Schloß

Entdecken Sie verborgene Türen und geheime Räume

Lösen Sie knifflige Rätsel und versteckte Reime

Stöbern Sie in der dunklen Vergangenheit des Schlosses

Erleben Sie grausame Ereignisse in düsteren Vollmondnächten

Vernichten Sie das Böse, bevor es Sie vernichtet

Handeln Sie nach den Regeln der Logik

Spielen Sie »Das Tier«

Den ganzen Horrorspaß gibt es für nur 69,90 DM*

für weitere Unterhaltung

RUFF and READY nur 20,— DM*

Für Amiga 500, 1000, 2000

MAB©91

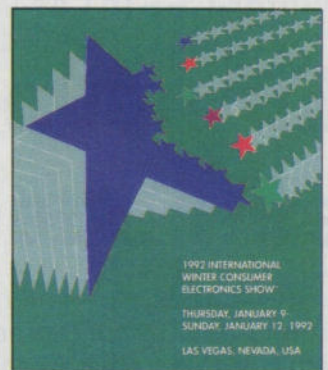
* Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland DM 4,— bzw. für das Ausland DM 6,— Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag • Postfach 250 • W-3440 Eschwege



AMIGA DOS Spieleteil 3/92

Neues aus Las Vegas



Einer der wichtigsten Termine, in den Kalendern fast aller international renommierten Softwarehäuser rot markiert, ist die Winter "Consumer Electronics Show" in Las Vegas. Die Stadt, sonst das Mekka der Glücksspieler, wird für vier Tage zum Podium, auf dem Computer und Software ein zentrales Thema sind. Insbesondere Spiele- und Unterhaltungssoftware sind die wesentlichen Themen der Messe. Zwar mußte sich der Amiga im "Land der unbegrenzten Möglichkeiten" mit harter Konkurrenz (PC und Spielkonsolen) auseinandersetzen, war aber trotzdem ein wichtiges Thema auf dieser Messe. Wir stellen Ihnen die interessantesten Neuheiten der Winter-CES vor. Welche Titel wird es für den Amiga geben? News und Termine der wichtigsten Elektronic-Show Amerikas. S.68

Kohle machen



Der Bergbau ist nicht totzukriegen. Eine neue Wirtschaftssimulation beschäftigt sich mit Förderung und Verkauf des "Schwarzen Goldes", und wir beschäftigen uns etwas näher mit dem Spiel. S.79

Flugsimulationen unter der Lupe

Drei neue Vertreter dieses beliebten Spielegenres warten darauf, Ihren Amiga in höchste Höhen zu entführen.

Flugsimulationen werden oft als der Sahneklecks auf dem großen Kuchen der Computerspiele bezeichnet. Kaum ein Softwarehaus, das nicht mindestens einen Vertreter dieses Genres im Sortiment führt. Microprose, Mirrorsoft und Electronic Arts verstärken die Riege der Computer-Flugmaschinen.



Was Sie in »Knights of the Sky«, »Flight of the Intruder« und »Birds of Prey« erwartet, können Sie in unserem Beitrag "Fliegen wie die Raubvögel" nachlesen. S.70

Bunt gemischt

In dieser Ausgabe von "Bunt gemischt" finden Sie **»ZACK«**, das Hütespiel, **»FANTASTIC VOYAGE«**, das Sie auf eine Reise in den menschlichen Körper entführt, den dritten Teil des Prügelepos **»DOUBLE DRAGON«**, der sich als nicht weniger rüde als seine Vorgänger erweist, **»SUSPICIOUS CARGO«**, ein lustig-unheimliches Weltraumabenteuer, und noch andere interessante Dinge mehr. Als besonderes Bonbon: Eine Preview über das erste Computerspiel an dem der **»Alien«**-Designer H. R. Gieger mitwirkte. S.80

Elvira

Das Busenwunder der Horror-Szene geht nun animiert auf Monsterjagd.

Nach zwei Adventures gibt es nun ein Arcade-Spiel um die Lady im schwarzen Abendkleid. Welche Untiere Sie dort erwarten und welche Aufgaben die gute Elvira diesmal erledigen muß, können Sie in unserer Review nachlesen. S.72



In den AMIGA-DOS Spieletips:

IMMORTAL
Überleben leichtgemacht! So gelangen Sie sicher ins zweite Level.

KINGS QUEST V
Sir Grahams Not hat ein Ende. Der erste Teil der Komplettlösung.

Alien Breed
Zeigen Sie es den Aliens. Tips zum Überleben und die wichtigsten Karten der Raumstation.

LETHAL XSSES
Nicht verzweifeln, jetzt kommt der AMIGA-DOS-Spezial-Cheatmode. S.75

Winter-CES in Las Vegas

Ein neues Spiel – ein neues Glück

Unwirtlich gab sich Amerikas Glücksspiel-Hauptstadt: Eisige Temperaturen trotz strahlender Sonne und ein im wahrsten Sinne des Wortes "umwerfender" Wind, der Fußgänger und Tumbleweeds unkontrolliert über die Straßen trieb, empfingen die Besucher aus aller Welt. Ein Grund mehr, daß man sich im warmen South Annex des Las Vegas Convention Center, in dem die Produkte der Unterhaltungssoftware zu begutachten waren, um so wohler fühlte.

Wohl fühlten sich dort auch die Hersteller dieses (besonders auf dem Konsolensektor mit immensen Zuwachsraten gesegneten) Industriezweiges. Keiner konnte sich über schlechte Geschäfte beklagen, auch wenn es immer wieder böse Bemerkungen über die Raubkopiererszene gab und viele neue Ideen, wie dem zu begegnen sei (allen voran OCEAN). Einziger großer Verlierer schien das CDTV-System, dessen Kontroversen mit dem (inzwischen von der Industrie favorisierten) CD-I inzwischen völlig in Vergessenheit geraten waren. Das interessanteste neue Medium jedoch war das CD-ROM, für das beinahe jedes größere Softwarehaus inzwischen Futter entwickelt.

Was auf der einen Seite erstaunte, auf der anderen Seite jedoch eigentlich doch nicht überraschend kam, war das gigantische Aufkommen an Konsolentiteln. *NINTENDO* hatte ein eigenes (wild im Winde wogendes) Riesenzelt aufgebaut, während der Stand von *SEGA* die Ausmaße einer kleinen "Stadt" inmitten der Halle einnahm. Die Meldungen über Rekord-Verkaufszahlen wurden jeden Tag von beiden Seiten in neue Höhen getrieben, und wer das Rennen letztendlich machen wird, steht in den Sternen.

Wie in den USA üblich, wo der Amiga-Markt nur einen verschwindend kleinen Anteil im Soft- und Hardwaregeschäft einnimmt, wurde die große Masse der neuen Programme zunächst für den PC hergestellt. Löbliche Ausnahmen bildeten dabei einige europäische Hersteller und das kanadische Softwarehaus *READY SOFT*, das in Europa unter dem EMPIRE-Label läuft.



Darklands: Ritterspiele im finsteren Mittelalter

Der Amiga-Markt in den USA

Traurig stimmte allerdings, daß etliche Hersteller auf Amiga-Umsetzungen ihrer PC-Produkte gänzlich verzichten wollen. In ihren Augen kann der Amiga, obwohl dessen Fähigkeiten noch selten wirklich ausgereizt wurden, mit den neuen PCs nicht mehr mithalten. Somit würde eine Umsetzung lediglich eine Verschlechterung des Programms bedeuten (sagt zum Beispiel *ORIGIN*), und das wolle man schließlich nicht.

Fast auf jedem Stand waren sie zu bewundern – Rollenspiele in 3D, Adventures mit phantastischen Szenarien, alles in allem neue und originelle Produkte. Mit großem Pomp wurde im kitschigen, übertrieben pseudomittelalterlichen Hotel EXCALIBUR das neueste Werk von *MICROPROSE* vorgestellt. Das Hotel bildete allerdings den passenden Rahmen für **»Darklands«**, wo man im Deutschland zur Ritterzeit, das heißt im 15. Jahrhundert, heldenmäßige Abenteuer bestehen darf. In Nürnberg nimmt das Spektakel seinen Anfang, Magie und Alchimie sind allgemein üblich, Raubritter, Hexen, Drachen und Diebe bevölkern das Land. Die ersten (PC-)Bilder konnten begeistern, eine Umsetzung für den Amiga steht allerdings leider noch nicht hundertprozentig fest.

LEGEND, ein Label das seit neuestem bei *ACCOLADE* erscheint, hatte mit der Fortsetzung seines Weltall-Epos **»Star Control II«** ein brandheißes Eisen im Feuer. Viel Wert wird hier auf Kommunikation mit Angehörigen fremder Kulturen, kurz Aliens, gelegt. Das fällt nicht weiter schwer, da besagte Aliens äußerst schnuckelig ausgefallen sind. Das Programm wird zwar im Juni zuerst auf dem PC erscheinen, Konvertierungen für den Amiga und das CDTV-System sind jedoch definitiv geplant. Kleines Schmankerl am Rande: Der echt tolle Sound wurde auf dem Amiga entwickelt!

Adventures und Rollenspiele

»Guy Spy and the Crystals of Armageddon« ist ein Action-Adventure, das aus dem Stall von *READY SOFT* stammt. Sehr schöne Vektorgrafiken bilden das Umfeld des Superagenten Guy Spy, der einen verrückten Wissenschaftler daran hindern muß, die für dessen geplante Vernichtungsmaschine benötigten Kristalle zu besorgen. Um die ganze Welt geht die abenteuerliche Jagd – allerdings erst im Herbst.



Hotel: Pseudomittelalterlich und hübsch kitschig

Ein Klassiker wird fortgesetzt: **»Eye of the Beholder 2«** erfreute das Auge (und den Rollenspielfan) bei *SSI*. Buck Rogers darf ebenfalls neue Abenteuer erleben in **»Matrix Cubed«** und auch **»Treasures of the Savage Frontier«** ist die Fortsetzung einer erfolgreichen Reihe. Neu dagegen und nicht im AD&D-Universum angesiedelt ist **»Tales of Magic«**, das mehr als 100 digitalisierte Grafi-

ken enthält, sehr ansprechend aussieht und genau wie die übrigen Programme noch in diesem Frühjahr erhältlich sein wird (PC und Amiga).

Qualität auf dem Vormarsch

Erfreulich war, was *ELECTRONIC ARTS*-Chef Simon Jeffery zu vermelden hatte: In Zukunft wird bei diesem Software-Riesen der Amiga wieder stärker gefördert (ganz im Gegensatz zu *ACTIVISION*, wo Amiga und Atari völlig in der Versenkung verschwinden). Ein wahrer Augenschmaus war bei *EA* das im Sierra-Stil ablaufende Adventure **»Sherlock Holmes«**, das seine Premiere im Juli auf dem PC und etwa drei Monate später auf dem Amiga haben soll. Im viktorianischen England angesiedelt, muß der Meisterdetektiv im Point-and-Click-Verfahren einen mysteriösen Mordfall aufklären. Ein sehr intelligentes Programm, das zudem mit sehr schöner Grafik und Sprachausgabe glänzte.

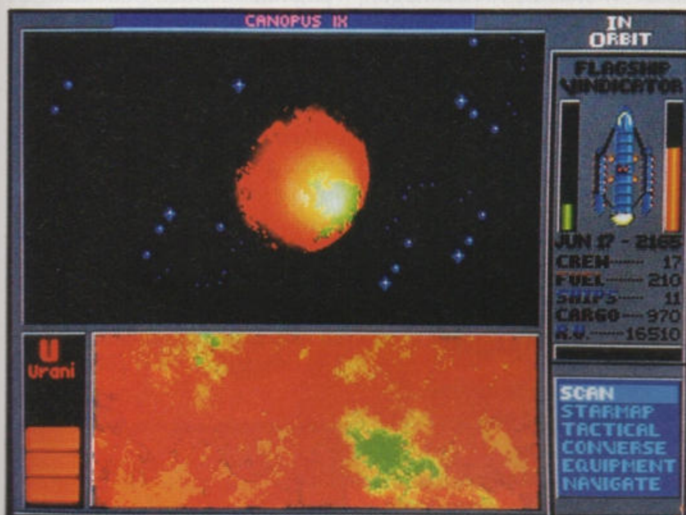
»Might & Magic« von *NEW WORLD COMPUTING* erhält reichlich Zuwachs. Zu sehen gab es bereits **»M&M III«**, Nummer 4 und 5 für den PC stehen ebenfalls kurz vor ihrer Vollendung (die Screens sahen echt super aus), die Amiga-Versionen werden noch etwas auf sich warten lassen. Die neueren M&Ms werden dann nicht mehr numeriert, sondern betitelt sein, damit die früheren Programme nicht ihre Attraktivität verlieren, war vom NWC-Boß zu hören.

Ein ganz besonderer Leckerbissen war von einem relativen Newcomer der Softwareszene zu sehen: **»Darkseed«** von *CYBERDREAMS*. Wer mehr über dieses animierte Horror-Abenteuer wissen möchte, sollte sich die Preview in dieser AMIGA-DOS-Ausgabe ansehen. Es lohnt sich!

Auch das in Europa von Gremlin vertriebene Softwarehaus *GAMETEK* wartete mit einem phantastischen Rollenspiel auf. **»Daemonsgate«**, auf der Sommer-CES bereits angekündigt, wird nun im März auf dem PC und im Sommer auf dem Amiga seinen Einstand geben. Interaktive Charaktere und ein ganzer Kontinent warten auf den Dämonenkiller. Probleme bei der Lösung?

Aber nicht doch! Dem Programm liegt eine Videocassette bei, auf der neben einer tollen Intro auch noch Hints und Tips enthalten sind. Das nenne ich Service ...

Auf dem Konsolensektor extrem erfolgreich, steigt *KONAMI* jetzt auch in das Computergeschäft ein. Mit **»Champions«** geht es gleich richtig zur Sache. Das Adventure, das von denselben Designern kreiert wird, die auch das gleichnamige Brettspiel entworfen haben, enthält sehr abwechslungsreiche Szenarien, ist leicht zu steuern und weist auch reichlich Action-Sequenzen auf. Die PC-Version soll im April, die Umsetzung für den Amiga im Herbst fertig sein.



StarControl II: Kosmisches Spielvergnügen



Indiana Jones and the Fate of Atlantis

Bei *LUCASARTS* bastelt man weiterhin fleißig an **»Indiana Jones and the Fate of Atlantis**, und auch bei *SIERRA* war außer Umsetzungen nicht viel in Sicht.

Ein recht witziges Spiel, das entfernt an die Strategieprobleme bei **»Lemmings«** erinnert, bringt *GAMETEK* mit **»The Humans»** ab April auf den Amiga. Steinzeitmännekens, die zu anfang lediglich menschliche Leitern bilden können, im Laufe der Levels aber immer mehr "technisiert" werden, müssen durch manchmal ganz schön schwierige Levels zum Ausgang gelotst werden. Ein Probespiel wurde zu einer Probeorgie ... Ähnlich verhält es sich bei den **»Gadget Twins«**, wo zwei niedliche fliegende großäugige Bälle, die mit allerhand Werkzeugen ausgestattet sind, zwölf Levels nach einem königlichen Diamanten durchsuchen müssen.

KONAMI zeigte auch bei den Action-Spielen recht Ansprechendes. Die berühmten **»Hero-Turtles - the Manhattan Mission«** räumten furios mit ihren Gegnern auf. Die 15 Missionen, die es vor dem großen Showdown zu absolvieren gilt, können netterweise in beliebiger Reihenfolge abgeprügelt werden.

Ballern, nein, Action, ja!

Gleich eine ganze Palette neuer Programme stellte *EMPIRE* vor. Neben Kriegsspielen wie dem im April erscheinenden **»Campaign«**, in dem man 120 Szenarien mit einem Panzer 3D-Vektor-mäßig unsicher machen kann, darf man bereits ab März mit dem »Team Yankee«-Nachfolger **»Pacific Islands«** fünf Südseeinseln von bösen Feinden befreien. Friedlicher geht es bei dem Olympia-infizierten **»International Sports Challenge«** zu, wo fünf Sportarten inklusive Schwimmen, Turmspringen und Reiten zu absolvieren sind.

Die Steuerung ist etwas gewöhnungsbedürftig, das Programm jedoch ganz witzig. Nicht witzig, sondern eher seltsam ist ein im März erscheinendes Programm mit Titel **»Eye of the Storm«**. Der Spieler hat hier den Job eines Raumschiff-Kameramanns übernommen, der für das nur noch life sendende Fernsehen spannende Bilder einfangen muß. Singende Vektorwale oder Raumpiraten – egal, Hauptsache die Kohle stimmt. ❑

Antje Hink (bs)



Über den Wolken

Fliegen wie die Raubvögel

Flugsimulatoren erfreuen sich sowohl bei den Computeranwendern als auch bei Programmierern größter Beliebtheit. Obwohl heute schon selbstverständlich, hat der uralte Menschheitstraum vom Fliegen nur wenig von seinem Reiz verloren. Doch ist es inzwischen nicht mehr damit getan, sich einfach in die Wolken aufzuschwingen. Der gute alte »Microsoft Flugsimulator II«, ein absoluter Klassiker dieses Spielegenres, in dem mit einer kleinen Sportmaschine der Luftraum über Amerika unsicher gemacht werden durfte, lockt kaum noch jemanden hinter dem Ofen hervor. Denn geflogen ist heutzutage fast jeder schon einmal, und sei es nur nach Mallorca.

Flugsimulationen haben sich, mit einigen Ausnahmen, auf die Umsetzung von Themen verlegt, die den meisten Menschen nicht so einfach zugänglich sind wie ein Flug in die Ferien. Düsenjets, Langstreckenbomber und Doppeldecker sind solche exotische Maschinen, die zumeist nur von unten im Vorbeiflug oder aber in der Tagesschau zu sehen sind. Drei neue Simulatoren haben sich eben solcher Flugzeuge angenommen und sie zu ihrem Thema gemacht.

Knights of the Sky

Während des Ersten Weltkrieges tauchten ihre Silhouetten zum erstenmal am Himmel über Flandern und Nordfrankreich auf: Die Doppeldecker, mit Fug und Recht als Legenden der Luftfahrtgeschichte bezeichnet, revolutionierten die Kriegsführung. Zwar war ihre militärische Bedeutung noch gering, doch zeichnete sich bereits ab, welch wichtige Rolle die Fliegerei in Zukunft spielen würde. Damals jedoch galten Piloten noch als die Ritter der Lüfte, und jeder Flieger versuchte, das "As der Asse" zu werden. »Knights of the Sky« verwandelt Ihren Amiga in einen tuckernden Doppeldecker und Ihre Maus (oder den Joystick) in einen Steuerknüppel. Diese technisch versiert programmierte Simulation ermöglich es dem Amiganer, sich mit verschiedenen Zielen in den Himmel über Flandern aufschwingen: Er kann auf einem von 20 authentischen Flugzeugen jener Tage trainieren, sich mit feindlichen Fliegerassen ein Luftkampfduell liefern oder in einem geschichtlich korrekt recherchierten Kriegsszenario diverse Missionen erledigen. So nebenbei gilt es, in der letzten Spielvariante Orden für besonders gut vollbrachte Aufträge einzuheimsen, möglichst viele Luftkämpfe zu überleben und dadurch vielleicht einmal zum "As der Asse" zu werden.

Gute Grafik, stimmige Sounduntermalung und diverse technische Gimmicks, wie zum Beispiel eine "Wiederholungs-Funktion", die bestimmte Szenen des eigenen Fluges aus frei wählbaren Blickwinkeln erneut zeigt, oder eine "Modem-Option", die es erlaubt, via Telefon gegen einen anderen Amiganer heiße Luftkämpfe zu bestehen, runden »Knights of the Sky« ab.



Es stehen mehrere historische Maschinen zur Auswahl

Flight of the Intruder

Wie bei den eben besprochenen "Rittern der Lüfte", so geht es auch hier ums Fliegen vor dem Hintergrund eines realen, glücklicherweise beendeten, Krieges. Damit hören die Ähnlichkeiten jedoch schon auf, denn in der Tat haben ein Doppeldecker und eine A-6 Intruder soviel gemeinsam wie ein Go-Kart und ein Porsche.

Als Hintergrund wurde der Vietnamkrieg gewählt. Der Spieler darf in dreizehn verschiedenen Missionen seine Fähigkeiten als Kampfflieger unter Beweis stellen. Alle Missionen basieren auf militärischen Aktionen, die im Rahmen der "Linebacker"-Offensive tatsächlich stattgefunden haben.

Neben der schon erwähnten A-6 steht noch die legendäre F-9 Phantom als Kampfflugzeug zur Auswahl. Als Basis für die Missionen dient ein Flugzeugträger, dessen Brücke auch die zentrale Einsatzleitstelle für die Missionen darstellt. Von dort erhält der Spieler über Funk laufend wichtige Informationen über beispielsweise die Wetterverhältnisse oder bedeutsame Feindbewegungen. Missionen werden bei »Flight of the Intruder« nicht von einer einzelnen Maschine geflogen, jedoch braucht nicht die komplette Staffel gesteuert zu werden. Der Spieler kann aber, so er möchte, zwischen einzelnen Maschinen einer Staffel hin- und herschalten.

Bei den dreizehn Missionen ist im wesentlichen immer nur das Ziel des Auftrages vorgegeben.



Doppeldeckerpilot im Ersten Weltkrieg

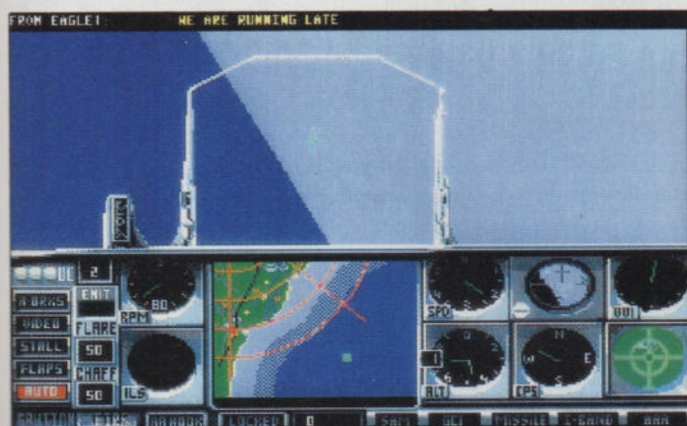
Mit steigender Flugerfahrung kann der Ablauf der Mission, Anflugsweg, Bewaffnung, Art der Flugformation und Startposition, selbst geplant werden. »Flight of the Intruder« vermittelt ein wenig wehrtechnische Historie und enthält viele kriegsgeschichtliche Details dieses Konfliktes, dessen Nachwirkungen noch heute von größter Aktualität sind.

Birds of Prey

Der wildeste Vertreter der Simulationen, die hier vorgestellt


Die Bedienung der Maschinen ist eine Wissenschaft für sich

werden, ist »Birds of Prey«, was soviel wie "Raubvögel" bedeutet. Und Raubvögel sind die 40 Maschinen allesamt, die den zentralen Gegenstand dieser Simulation bilden. Auch dieses Programm ist wieder vor einem außerordentlich kriegerischen, diesmal allerdings fiktiven Hintergrund angesiedelt. Es gilt,



Im Cockpit der F-9 Phantom: Technik bis in die letzte Ecke

mit einem umfangreichen Arsenal, das 26 verschiedene Flugzeugtypen (vom Aufklärer bis zum Bomber) enthält, eine Miniversion eines "Dritten Weltkriegs" zu führen, bei dem keine Nuklearwaffen zum Einsatz kommen. Das Arsenal der Gegenseite ist nicht minder umfangreich und furchterregend: 13 Flugzeuge aus sowjetischen Waffenkammern machen den Gegner, der ausdrücklich nicht als Warschauer Pakt, sondern als eine fiktive Macht mit allerdings sowjetischer Waffentechnologie bezeichnet wird, zu einer echten Herausforderung der strategischen und fliegerischen Fertigkeiten des Spielers.

Neben den Flugzeugen verfügt jede Seite noch über ein bestimmtes Kontingent an Flughäfen, Munitionsfabriken, Radaranlagen und anderen wichtigen Einrichtungen. In Flugeinsätzen, bei denen der Spieler die Maschine auswählen kann, die ihm dafür am besten geeignet erscheint, müssen strategisch wichtige Ziele auf feindlichem Territorium angeflogen und vernichtet werden.

Die gegnerische Luftabwehr schläft natürlich nicht und versucht, diese Einsätze zu vereiteln. Neben der Finesse, die bei der Steuerung des Flugzeugs im Luftkampf vom Spieler erwartet wird, bedarf es auch einiger Überlegung bei der Auswahl der Ziele: Gelingt es, gegnerische Radaranlagen zu vernichten, wird dieser dadurch sozusagen "blind" und kann den Angriffen des Spielers erst sehr spät durch Abfangjäger begegnen. Zerstörte Munitionsfabriken können nicht mehr produzieren, und der "Feind" ist auf bestehende Vorräte angewiesen. Von Flughäfen mit zerstörten Rollfeldern kann kein Flugzeug mehr starten.


Planung der "Intruder"-Einsätze



Technische Details mit Bewaffnungs-Diagramm

Die Programmierer von Argonaut Software haben ganze Arbeit geleistet: Jede Maschine weist individuell verschiedene Flugeigenschaften auf, Waffensysteme und technisches Zubehör wurden genau recherchiert und strategische Grundtaktiken der High-Tech-Kriegsführung detailgetreu in die Simulation übertragen.

Fazit

Bis auf die nostalgisch angehauchten »Knights of the Sky« sind die hier vorgestellten Programme echte Simulations-Schwergewichte. Zu »Birds of Prey« und »Flight of the Intruder« erhält der Spieler jeweils eine umfangreiche deutsche Bedienungsanleitung (das Manual von »Knights of the Sky« ist leider nur in englischer Sprache vorhanden), die sich beide nicht mit den Auflistungen technischer Details zufriedengeben. Neben Kapiteln, die einen Schnelleinstieg erleichtern, finden sich auch genaueste Beschreibungen aller Spielmechanismen, was angesichts der komplexen Abläufe auch nötig ist.

Allen drei Spielen ist gemeinsam, daß hier Langzeitunterhaltung geboten wird. Dementsprechend sind selbstverständlich auch Speicheroptionen vorhanden. Erwartet den Spieler bei »Birds« und »Intruder« echte "Hardcore"-Simulation, fallen die "Ritter der Lüfte" in bezug auf die Komplexität etwas ab. Hier wurde weniger auf High-Tech-Simulation denn auf schnellen Einstieg und leicht erschließbaren Spielspaß Wert gelegt. Der ist in den anderen beiden Programmen auch enthalten, hier bedarf es jedoch erst einiger Anläufe, bis alles verstanden wurde und tadellose Flugergebnisse erzielt werden.

Benutzerfreundlich angelegt, lassen sich von allen drei Programmen recht problemlos Sicherheitskopien anfertigen, die umfangreichen Handbücher erweisen sich als erstklassiger Kopierschutz, denn ohne die ist man ziemlich aufgeschmissen. »Birds of Prey« und »Knights of the Sky« sind außerdem auch auf Festplatte installierbar; eine Option, die »Flight of the Intruder« leider vermissen läßt. ❑

(hs)

Knights of the Sky
Hersteller: Microprose
Muster von: Microprose
Preis: 99,95 DM
Konfiguration lt. Händler: A500, 1000, 2000, 3000, 1 MByte

Flight of the Intruder
Hersteller: Mirrorsoft
Muster von: Mirrorsoft
Preis: 119,95 DM
Konfiguration lt. Händler: A500, 1000, 2000

Birds of Prey
Hersteller: Electronic Arts
Muster von: Electronic Arts
Preis: 99,95 DM
Konfiguration lt. Händler: A500, 1000, 2000, 2500, 3000, 1 MByte



Elviras Spezialität – Eiswüstenerforschung im Negligé

Elvira – The Arcade Game

Die Lady mit der schwarzen Robe und den großen Ohren ist wieder da. Nachdem sie in diversen Adventures dem Bösen die Stirn bot, erweitert sie ihre Karriere in diesem Spiel noch und zeigt Profil als Lauf- und Springheldin, die obendrein mit diversen Zaubereien allerlei Untiere aus der Welt schafft.

Die ganze Geschichte beginnt mit dem Auftauchen des Geistes König Gustavs von Transsylvanien, der Elvira mitteilt, daß sie Erbin seiner Krone ist. Elvira, die natürlich größten Wert darauf legt, ihr Erbe anzutreten, muß jedoch erfahren, daß sie zunächst königliche Qualitäten unter Beweis stellen muß, indem sie die beiden Teile von König Gustavs Reich erobert. Also macht sich die Dame mit dem wogenden Busen auf, um die "arktische Urwelt" und das "feurige Höhlenreich" zu gewinnen.

Laufend, hüpfend und Zaubersprüche schleudernd, wandert Elvira durch die in der Seitenansicht dargestellten Schauplätze des Spiels, sammelt dabei diverse Extras ein und versucht, den Ausgang aus den umfangreichen Szenarien zu finden. Das Gelände gleicht einem Labyrinth, ist durchsetzt mit Plattformen und Höhlen, die es zu erkunden gilt.

Obwohl rundum sauber programmiert, mit netter Grafik und gutem Sound versehen, kommt »Elvira – The Arcade Game« über reines Mittelmaß nicht hinaus. Schade eigentlich ... ❑

(hs)

Elvira

AMIGA DOS

Testurteil 3'92

befriedigend 70%

Grafik:	70%
Sound:	65%
Handling:	65%
Spielspaß:	65%
Preis/Leistung:	65%

Hersteller: Flair
Muster von: Leisure Soft
Preis: ca. 79,95 DM
Konfiguration lt. Händler: Amiga 500, 1000, 2000; 1 MByte

Fuzzball

Der Zauberlehrling kann es nicht lassen! Kaum ist der Meister abwesend, nimmt er dessen Zauberstab und öffnet die geheime Zaubertruhe. Diese ist eine Miniversion von Pandoras Büchse und setzt böse kleine Fellbällchen (Fuzzballs) frei, die sich wie eine Lawine über die fünfzig Kammern des Turmes ausbreiten, in dem Zauberer und Lehrling hausen. Der letzte Versuch, zu retten, was noch zu retten ist, geht obendrein schief, und der arme Lehrling verwandelt sich selbst in einen Fellball.

Der Meister, ob der ganzen Geschichte sehr erbost, weigert sich, dem armen Schüler wieder seine richtige Gestalt zurückzugeben, bevor dieser nicht alle Kammern des Turmes von der Fellball-Invasion befreit hat. Die einzelnen Turmzimmer stellen die Levels dar, durch die Sie den verzauberten Lehrling steuern müssen. Der muß sich bei seinen Exkursionen jedoch vor den "Fuzzballs" in acht nehmen, die jetzt in jeder Kammer herumlaufen und nur darauf warten, den Zauberlehrling in einen der überall klaffenden Abgründe zu schubsen. Wehrlos ist der arme Schüler jedoch nicht ganz, kann er doch kleine Bällchen auf die Gegner spucken, die sich, sobald sie oft genug getroffen wurden, zusammenballen und ein-, zwei-, dreimal auf und niederhüpfen und währenddessen, so wie sie es gerne mit dem armen Lehrling tun, von den Plattformen gestoßen werden können. Überall in den Turmkammern liegen Kristallsplitter, Urnen, Juwelen und andere Wertgegenstände herum, die eingesammelt werden müssen. Erst wenn es gelungen ist, alles mitzunehmen, geht es in der nächsten Kammer weiter. ❑



Fellbällchen überall und der Zauberlehrling mittendrin

(hs)

Fuzzball

AMIGA DOS

Testurteil 3'92

gut 75%

Grafik:	75%
Sound:	65%
Handling:	65%
Spielspaß:	75%
Preis/Leistung:	75%

Hersteller: System 3
Muster von: System 3
Preis: ca. 59,95 DM
Konfiguration lt. Händler: Amiga 500, 1000, 2000

Volfied

In den grauen Anfängen der Computerspiele gab es ein Spiel namens »Styx«. Dabei ging es darum, ein Irrlicht, das auf einer Fläche umherflimmert, einzusperren, indem die Fläche verkleinert wurde. Mittels eines Cursor-Stiftes konnten Rechtecke in der Fläche eingezeichnet werden, auf die das Irrlicht nicht mehr wandern konnte. Gelang es, mehr als 80 Prozent vom Tummelplatz des Lichtes mit dem Cursor-Stift zu markieren, gab es einen reichlichen Punkte-Bonus. Die Fläche wurde wieder gesäubert, und der Spieler stand erneut dem Irrlicht gegenüber, das sich nun etwas schneller bewegte als bisher.

Im Zuge der Aufarbeitung älterer Computerspielideen kommt nun auch dieses Spiel, mit neuer, besserer Grafik und einigen Extras versehen, wieder auf den Markt. Das Irrlicht wurde durch verschiedene Roboter ersetzt, die sich nicht mehr damit begnügen, herumzufliegen und den Cursor-Stift des Spielers durch Berührung zu vernichten, sondern obendrein noch mit den tückischsten Waffen herumballern. Den mechanischen Gegnern stehen mehrere kleine Drohnen zur Seite, die wie ihre großen Kollegen dem Cursor-Stift nachstellen. Gelingt es, Extracontainer einzusammeln, die sich auch auf den Flächen befinden, gibt es vom "Speed-Up" bis zur Bordkanone die verschiedensten hilfreichen Extras.

Die gute Grundidee, die hinter diesem Spiel steckt, hat durch Neuüberarbeitung nur gewonnen. Hier sind zwar keine programmiertechnischen Meisterleistungen zu bewundern, aber dafür gibt es Spielspaß satt. ❑

(hs)



Kesseln Sie den Roboter ein

Volfied

AMIGA DOS

Testurteil 3'92

gut	80%
Grafik:	70%
Sound:	65%
Handling:	80%
Spielspaß:	85%
Preis/Leistung:	80%

Hersteller: Empire
Muster von: Empire
Preis: 79,95 DM
Konfiguration lt. Händler: Amiga 500, 1000, 2000

Leander

Prinzessinnen, Dämonen und tapfere Helden, das ist der Stoff, aus dem Legenden sind und der auch die Hauptingredienzien von »Leander« bildet.
Der Dämon hat die Prinzessin entführt, und Leander, der Held, versucht, sie zu befreien. Ein nicht gerade leichtes Unterfangen, bei dem der Tapfere dringend Hilfe benötigt, die ihm jeder geneigte Amiga-Benutzer angedeihen lassen kann. Laufen, Springen und Kämpfen, darum geht es im wesentlichen, und von daher gebühren Leander nicht gerade Preise für besonders originellen oder neuartigen Spielverlauf. Die Umsetzung altbekannter Spielmechanismen kann allerdings als beachtlich bezeichnet werden und verspricht nicht unbeträchtliches Vergnügen. Die Abenteuer des Helden führen den Spieler durch diverse Schauplätze, in welchen immer eine bestimmte Aufgabe erfüllt werden muß, um das Teleporter-Tor zum nächsten Schauplatz zu öffnen. Zumeist muß Leander etwas finden, das irgendwo im Szenario, in dem er sich gerade befindet, versteckt ist. Dies können besondere Juwelen, bestimmte Gegenstände wie beispielsweise ein spezieller Bogen oder einfach ein besonderer Schlüssel sein. Bei der Erledigung der Aufgaben versuchen die verschiedensten Bösewichter, Leander ins Handwerk zu pfuschen: Ballonfliegende Trolle, lanzenschwingende Söldner, Riesenspinnen, Drachen, Felsbrocken, wilde Hunde, Fledermäuse, sporenspuckende Killerpflanzen und viele andere Finsterlinge mehr haben es auf den Helden abgesehen. Der ist jedoch nicht ganz hilflos, denn neben seinen Sprungkünsten versteht er sich auch darauf, sein Schwert sprechen zu lassen.
Die Macher von Leander haben sich deutlich an der Benutzerfreundlichkeit japanischer Modulspiele orientiert, und das hat dem Spiel nicht geschadet. Daß erweiterter Speicher unterstützt wird, ist heute schon so gut wie selbstverständlich; das Optionsmenü, in dem der Spieler allerlei Einstellungen vornehmen kann, nicht. So ist es hier beispielsweise möglich, den Schwierigkeitsgrad des Spiels einzustellen, die Art des Joysticks, mit dem gespielt wird, kann ausgewählt werden, und es ist möglich, "Credits" zu aktivieren, durch die man nach dem Ende eines Spiels das nächste wieder an derselben Stelle beginnen kann.
Die grafische und die musikalische Präsentation des Spiels vermögen ebenfalls zu überzeugen, so daß Leander ein rundum gelungenes Spielvergnügen darstellt, das an sich nur empfohlen werden kann. ❑

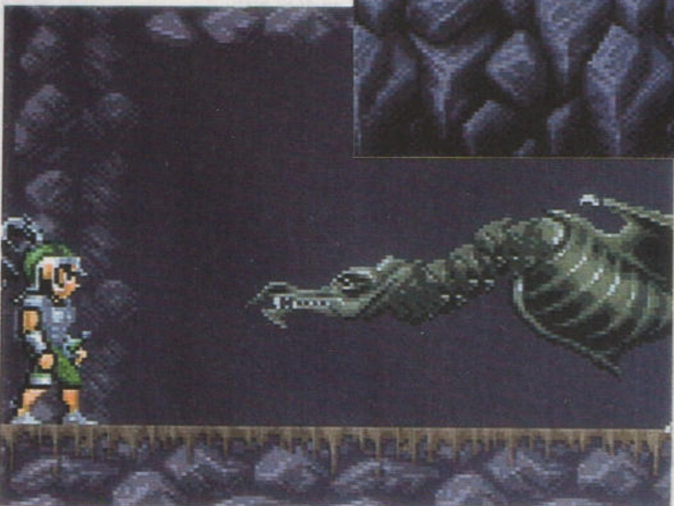
(bs)



Im Shop gibt es Rüstungen und bessere Waffen


Trolle...




... Spinnen und Drachen kreuzen Leanders Weg

Leander

AMIGA DOS

Testurteil 3'92

gut	**85%**
Grafik:	90%
Sound:	80%
Handling:	85%
Spielspaß:	90%
Preis/Leistung:	85%

Hersteller: Psygnosis
Muster von: Psygnosis
Preis: 89,90 DM
Konfiguration lt. Händler: 1 MByte



Willkommen in der Wunderwelt der Spieletips. Wie Sie sicherlich bemerkt haben werden, hat sich das Layout des Spieleteils ein bißchen geändert. Daß Sie hier Tips und Tricks, Rat und Tat und obendrein noch ein paar Cheats zu Spielen jeder Couleur erhalten, bleibt gleich. Daß diese Rubrik von Ihrer Mitarbeit lebt und daß es für feine Tips auch feine Prämien gibt, bleibt auch in Zukunft unverändert. Natürlich können wir keine Reichtümer ausschütten. Jedoch gibt es für jeden veröffentlichten Tip ein tolles Bonusspiel. Also, halten Sie mit Ihren Erfahrungen nicht hinter dem Berg, schreiben Sie uns ...

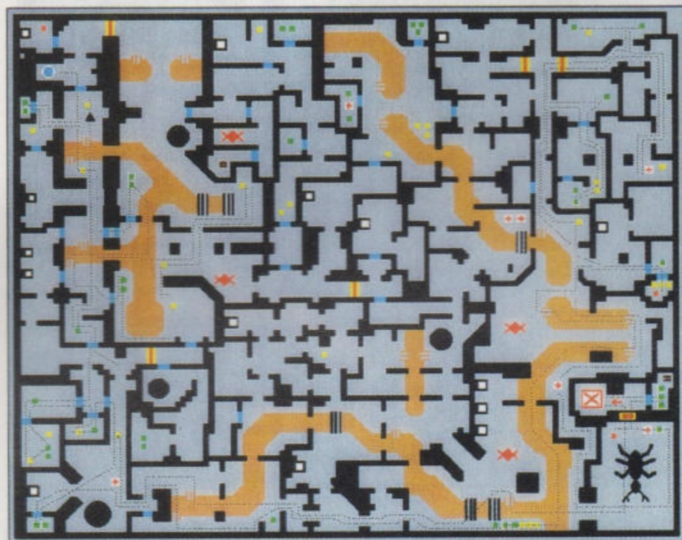
Alien Breed

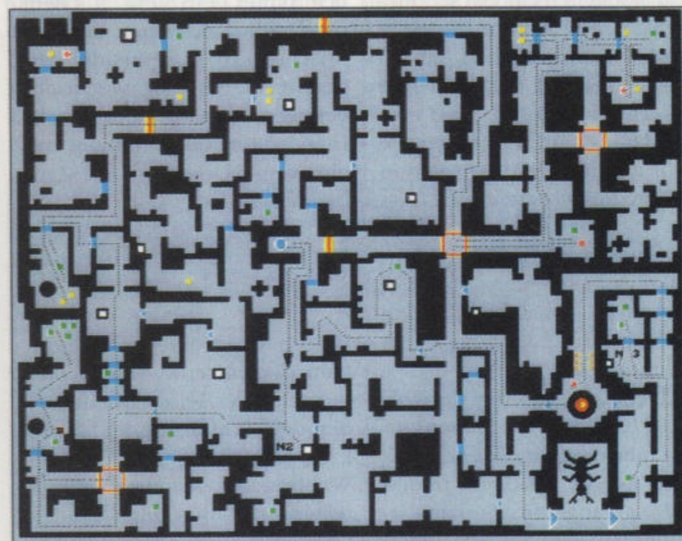
Als Hors d'oeuvre wollen wir Ihnen helfen, eine Bande fieser Außerirdischer am Kragen zu kriegen. Für »Alien Breed«, im letzten Heft erst getestet, gibt es heute schon Karten, die verraten, wie der bösen Brut garstiger Aliens am besten beizukommen ist. Vorweg jedoch erst ein paar Tips allgemeiner Natur.

Zunächst sollten Sie versuchen, "Key-Cards" zu sparen. Betreten Sie nicht sinnlos Räume, in denen nur eine 100-Credit-Banknote liegt. Bleiben Sie ständig in Bewegung; es tauchen immer neue Aliens auf, wenn Sie zu lange an einer Stelle stehen bleiben. Nicht zu nahe an Türen vorbeigehen, die Sie nicht zu öffnen gedenken, da es leicht passieren kann, daß Sie der Tür zu nahe kommen, sie sich öffnet und eine wertvolle "Key-Card" vergeudet wurde. Mit den schweren Feuerschutztüren vorsichtig sein. Sie sollten versuchen, sie nicht zu schließen, denn so kann es passieren, daß man sich einsperrt, ohne es zu wollen.

Level eins erledigt sich sozusagen fast von allein und bedarf daher keiner weiteren Erklärungen. Jedoch schon im zweiten Stockwerk der mit Aliens durch und durch infizierten Raumstation kann man sich ganz leicht verlaufen. Folgen Sie jedoch dem Weg, der auf den Karten durch eine gepunktete Linie dargestellt ist, dann kann im Grunde nichts mehr passieren.


Alien Breed, Level 2

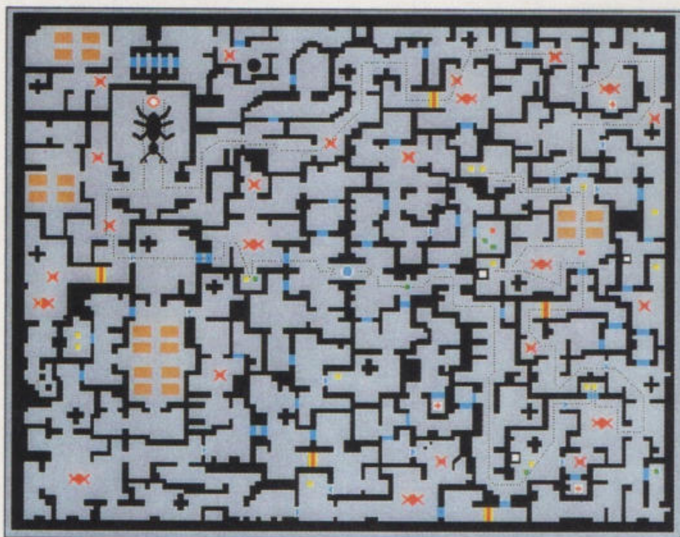

Alien Breed, Level 3

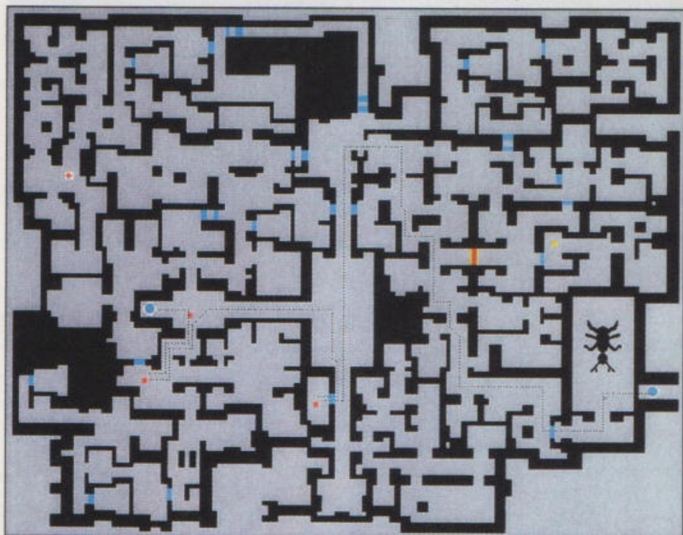

Alien Breed, Level 4

Kings Quest V

Ganz so einfach scheint die Suche nach den verschleppten Familienangehörigen des lieben Sir Graham doch nicht zu sein. Immer wieder erreichen uns Anfragen zu einigen kniffligen Stellen des Spiels. Ihre Wünsche sind uns Befehl, und so finden Sie hier den ersten Teil einer Reihe von Tips zu Kings Quest V.

Beim Ameisenbau: Wenn der Hund das arme Völkchen


Alien Breed, Level 5


Alien Breed, Level 6

malträtiert, können Sie ihm mit dem Stöckchen ein viel schöneres Spielzeug geben. Die Hilfe der Ameisen brauchen Sie bei der Suche nach der goldenen Nadel im Heuhaufen.

Der Bär, der ein Bild weiter die nicht minder bedauernswerten Bienen piesackt, zeigt sich über den Fisch (der in der Tonne in der Stadt fröhlich vor sich hinstinkt) sehr angetan. Die Bienenkönigin verschenkt eine Honigwabe, die später noch an Bedeutung gewinnt.

Nach einigen Toden in der Wüste (die Schlauen schauen auf der hier abgedruckten Karte nach, wo die Oasen zu finden sind!) gelangt Sir Graham irgendwann einmal zu dem großen Felsenportal. Zunächst versteckt er sich, ein Bild südlich, hinter dem Stein, aus dem die Quelle entspringt, und beobachtet von dort aus die beiden Reiter. Sind die wieder abgezogen, folgt man ihnen in ihr Lager. Dort kann Graham mit ein wenig Geschick den Zauberstab stehlen.

Zurück am Felsenportal, wird er in der beobachteten Art und Weise eingesetzt (vorher abspeichern).

Im Schatzgewölbe nimmt Graham die einzelne Münze und die reich verzierte Flasche neben dem Portal. Mit der Münze kann der Prinz jetzt zu der Wahrsagerin Madame Muska gehen, die ihm ein Amulett übergibt.

Mit dem Amulett und der Flasche aus der Schatzhöhle bewaffnet, gehen des Prinzen Abenteuer im Hexenwald weiter. Das Amulett schützt vor dem bösen Zauber der Hexe, und die Flasche (beziehungsweise der darin enthaltene Flaschengeist) sorgt für einen stilvollen Abgang der bösen Braut. So, in der nächsten AMIGA DOS werden wir Ihnen erklären, wie man an der Klapperschlange vorbeikommt und was in der Stadt alles erledigt werden muß. Bis dahin können Sie ja ein wenig auf eigene Faust weiterknobeln.

Lethal Xsess

Eclipses Ballerepos ist kaum auf dem Markt, da trudelt auch schon ein echter Cheatmode ein, der das Überleben in diesem knüppelharten Ballerspiel ein wenig leichter macht. Um den Cheat zu aktivieren, brauchen Sie nur im Titelbild "COKE" einzugeben

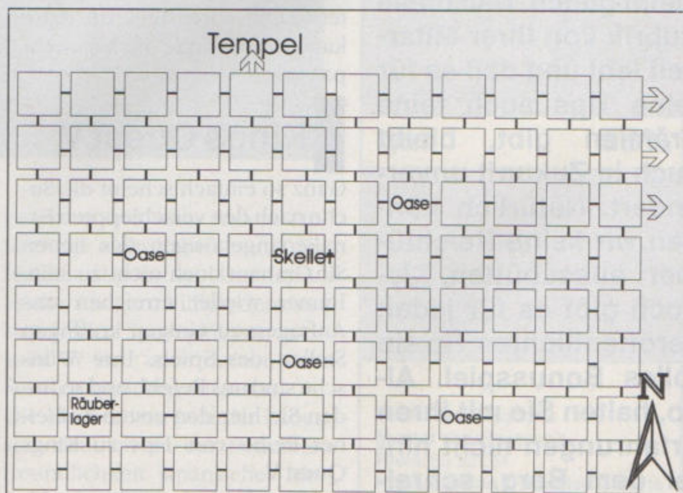
Elf

Elfen leben im Zauberwald gefährlich. Jedoch kann die Karte zum ersten Level, die Ihr auf diesen Seiten findet, sicherlich so manchen Irrweg ersparen. Dazu noch folgende Tips: Dem Burschen auf dem "stillen Örtchen" fehlt Papier. Der Vogel ist hungrig und mag für sein Leben gern Samenkörner. Ungebratene Hühnchen kann man über einem entzündeten Lagerfeuer bestens garen. Soviel zum ersten Level.

The Immortal

Verliese und unterirdische Bösewicht-Bastionen sind "in". Eines der tollsten Fantasy-Abenteuer dieser Machart ist »The Immortal«, berühmt für die "tausend tollen Tode", die man hier erleiden kann. Wie Sie hier weiterkommen, ohne dauernd zu sterben, wollten Sie wissen? Voilà, einfach weiterlesen!

Das erste Stockwerk ist noch recht einfach zu meistern. Zunächst muß der Erschlagene im ersten Raum durchsucht werden. Von den drei Feuerbällen, die Ihr Held hier finden kann, sollte er sogleich einen in Richtung des anstürmenden Goblin abfeuern. Der zweite Feuerball ist für den nächsten Goblin bestimmt, der einen Raum weiter links gegen den einsamen Menschen kämpft. Der bedankt sich für die Hilfe mit einem Schlüssel, der an der Truhe im selben Raum das Schloß öffnet. Weiter geht es nach rechts und dann nach oben. Der dort postierte Goblin bewacht einen Haufen Goldstücke, die Sie beziehungsweise Ihr Held sehr gut gebrauchen können. Also, "Feuer frei". Sie wenden sich nach oben und gehen im folgenden Raum rechts herum dichtgedrängt an der Wand entlang. Auf diesem Weg entgehen Sie den meisten Pfeilen. Im nun folgenden kleinen



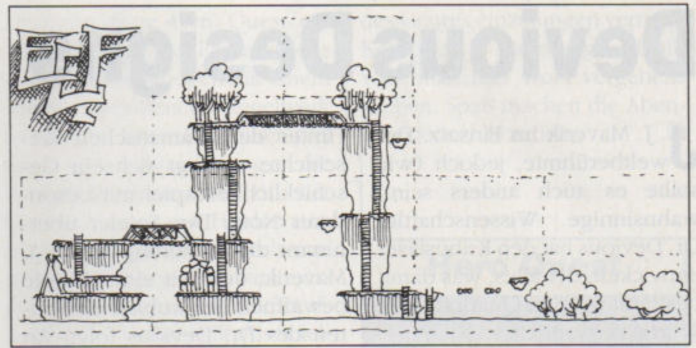
Kings Quest V - die Wüste

Zimmer lauern Fallgruben, die Sie durch erneutes An-der-Wand-Entlangschleichen vermeiden können. Der nächste Raum wird haarig, hier begegnet Ihr Held erstmals den Schatten, unsichtbaren, aber besonders fiesen Bösewichtern. Obendrein ist auch hier der Boden mit Fallgruben gespickt. Führen Sie Ihren Helden am oberen Rand des Raumes bis zur linken Wand; von dort einen schnellen Abstecher zu den blinkenden Gegenständen machen und diese (ein "Charm" und ein Amulett) aufsammeln. Schnell an der linken Wand weiter nach unten und fix durch die Tür schlüpfen. Jetzt können Sie Ihren Helden erst einmal ausruhen lassen. Danach benutzen Sie das Amulett, um den einfallenden Lichtstrahl zu reflektieren, und schon öffnet sich die Falltür zum nächsttieferen Stockwerk.

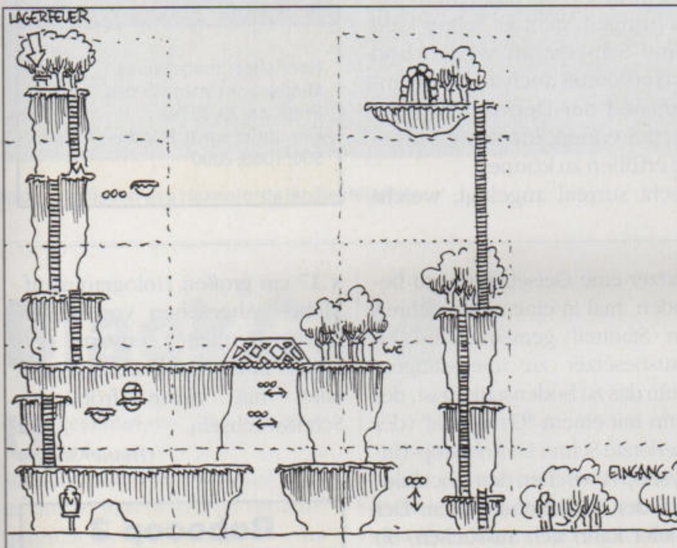
Gleich im ersten Raum ist Eile geboten, wenn Sie nicht wollen, daß Ihr Held zu einer saftigen Amöbenmahlzeit wird. Also, schnell zu dem roten Fleck am Boden, das Juwel, das dieser darstellt, wird eingesammelt, der Haufen Knochen durchsucht, und weiter geht's durch die nördliche Tür. Dort erwartet Sie schon ein Händler, der Ihnen einen "Anti-Amöben-Saft" verkaufen will. Sie schlagen sein erstes Angebot aus und fragen dann ein zweitesmal. Nun wird er billiger, und Sie können Ihrem Helden das Gebräu gönnen. Weiter geht es in Richtung Norden und nach links durch die Tür. Sobald die Waldlichter auftauchen, lassen Sie Ihren Helden das "Charm" benutzen. Hinter der Tür locken Sie den oberen der beiden Goblins, die die Tür bewachen, um die nördliche Ecke des Raumes und verschwinden wieder durch die Tür, bevor er Sie erwischen kann. Wieder in dem Raum angelangt, in dem auch der Händler sitzt, wenden Sie Ihren Helden jetzt der rechten Eisentüre zu. Dahinter liegt der Raum, in dem Sie eben schon einmal waren, nur bewacht jetzt nur noch ein Goblin die Tür, da Sie den anderen bereits weggelockt haben. Den Burschen kann Ihr Held nun recht einfach niederringen. Im nun folgenden Raum sitzen der Goblinkönig und zwei Wachen. In dem Strohhaufen werden die Sporen gepflanzt. Versuchen Sie, dies möglichst so zu bewerkstelligen, daß keine Wache Sie bemerkt. Ansonsten müssen Sie einen Kampf führen, den Sie Ihrem Helden eigentlich ersparen können. Die Sporen wachsen recht schnell zu giftigen Fliegenpilzen heran, die die Goblins töten (Ihren Helden auch, wenn er nicht schnell genug den Raum verläßt). Wenn Sie den Raum wenig später wieder betreten, sind die Wachen tot, und der König liegt darnieder. Wenn Sie Ihren Helden nun mit dem König sprechen lassen, wünscht dieser sich nichts sehnlicher als Wasser, für das er Informationen verspricht. Die Tips, die der Sterbende zu geben hat, merken Sie sich, und der Schlüssel wird mitgenommen. Jetzt geht es wieder zurück zu dem Raum, in dem grüne Amöben ihr Unwesen treiben. Bevor Sie den Raum jedoch betreten, lassen Sie Ihren Helden am besten erst einmal das Öl des Händlers auf die Stiefel reiben. Danach können Sie sich gefahrlos in den Raum begeben. Legen Sie dort den Stein auf den Boden, die Schleimwesen werden darübergleiten und mit ihren Säften den Stein wegätzen, so das der darin verborgene Edelstein zum Vorschein kommt. Verlassen Sie den Raum durch die Türe, die Sie bisher noch nicht benutzt haben. In der Ecke des nächsten Raumes sitzt ein Zwerg, der an einem weiteren Edelstein herumpoliert. Schütten Sie das Säckchen mit Staub (Dust) über ihm aus. Ein plötzlicher Sinneswandel ist die Folge, und der bisher griesgrämige Gnom übergibt nun bereitwillig das Juwel.

Nun haben Sie die drei Edelsteine beisammen, die benötigt werden, um den Zugang zum nächsten Stockwerk zu öffnen. Mit dem Schlüssel des Königs wird die versperrte Tür geöffnet, und wenn Sie Ihren Helden nicht auf dem Stroh ausruhen ließen, sollte auch noch genug von dem Anti-Amöben-Öl an den Heldenstiefeln kleben, um den nächsten schleimverseuchten Korridor gefahrlos zu passieren. Im nächsten Raum befinden sich drei Symbole, die in den Boden eingemeißelt wurden. Lassen Sie den Zauberer die gefundenen Edelsteine einsetzen, wie in der Skizze auf dieser Seite eingezeichnet. Nun öffnet sich im Boden der Zugang zum nächsten Level des Verlieses, dem wir uns in der nächsten Ausgabe der AMIGA DOS ausgiebig widmen werden. Bis dahin wünschen wir viel Spaß beim Ausprobieren dieser Tips. ❑

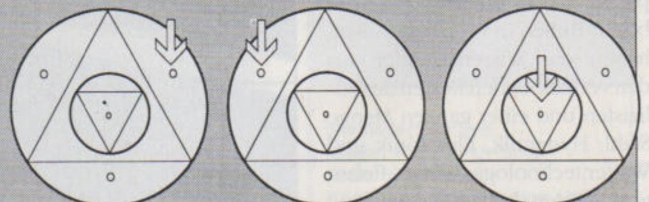
(bs)



Elf, der erste Teil des ersten Levels



Elf, der zweite Teil des ersten Levels



Immortal, die Symbole im zweiten Level

Devious Designs

J. J. Maverik im Einsatz. Der weltberühmte, jedoch (wie sollte es auch anders sein!) wahnsinnige Wissenschaftler Dr. Devious hat den Kubuslaser entwickelt, der alles, was damit bestrahlt wird, in Quadrate und Würfel verwandelt.

Hinter der dramatischen Geschichte verbirgt sich ein Geschicklichkeitsspiel mit besonderer Note. Der Spieler übernimmt die Steuerung von J. J. Maverik, der, mit einer Pistole bewaffnet, den kubischen Spuren des Dr. Devious folgt. An den in der Seitenansicht dargestellten Schauplätzen, an denen der Doktor wirkte, müssen aus herumliegenden Würfeln bestimmte Gebilde zusammengesetzt werden. Diverse Monster behindern die Arbeit des Agenten, können jedoch mit der schon erwähnten Pistole unschädlich gemacht werden. Neben den Untieren und einem knapp bemessenen Zeitlimit gibt es aber auch angenehme Überraschungen. So regnet es beispielsweise vom Fliegenpilz über verschiedene Fahnen bis zu kleinen Abbildern von J. J. Maverik selbst alle möglichen Gegenstände, die eingesammelt werden können.



J. J. Maverik beim Würfelpuzzle

Hat J. J. in den ersten Bildern schon alle Hände voll zu tun, die wild verteilten Würfel wieder in ihre ursprüngliche Form zu bringen, geht es schon bald ohne Schwerkraft weiter, und Maverik muß auch an den Wänden und der Decke der Bilder herumrennen, um seine Aufgabe erfüllen zu können.

Recht surreal angelegt, weicht »Devious Designs« vom üblichen Strickmuster, das Jump-'n'-Run- und Geschicklichkeitsspielen zugrundeliegt, ein ganz schönes Stück ab. ❑

(Heiner Stiller)

Devious Designs	
AMIGA DOS Testurteil 3'92	
gut	75%
Grafik:	75%
Sound:	60%
Handling:	70%
Spielspaß:	80%
Preis/Leistung:	80%

Hersteller: Image Works
Muster von: Image Works
Preis: ca. 79,95 DM
Konfiguration lt. Händler: Amiga 500, 1000, 2000

Robocop 3

Die Verbrechensbekämpfung der Zukunft geht in die dritte Runde, sowohl im Kino als auch auf Diskette. Die Stadt Central City wird nach wie vor von Kriminalität gepeinigt. Jeder dritte Stadtbewohner ist ein Schläger, jeder vierte ein Dealer und jeder fünfte ein Massenmörder; kurz, die Cops (amerikanischer Slangausdruck für Polizisten) haben nicht mehr viel zu melden und, dank der permanenten Unterbezahlung, auch wenig Lust auf ihren Job. Der Einzige, der nach wie vor mit ganzem Herzen, pardon ganzer Batterie bei der Sache ist, ist Officer Murpy, der Robocop. Einstmals von bösen Buben in Fetzen geballert, haben die Wissenschaftler aus den verbleibenden Resten des Polizisten und einer ganzen Menge Stahl, Hydraulik, Elektronik und Waffentechnologie einen Roboterpolizisten gebastelt, der nun auf den Straßen der Stadt patrouilliert und mit der Präzision eines Computers Kriminellen die Ohren vom Kopf schießt. Die Rolle des Stahlpolizisten darf der Spieler übernehmen. Wie des großen Vorbilds, so ist auch des kleinen Computercops Aufgabe das Unschädlichmachen diverser Krimineller, was fast immer mit Hilfe einer großkalibrigen Pistole geschieht. Wild um sich ballernd, geht es so durch mehrere 3D-Szenarien, die in flüssig animierter Vektorgrafik dargestellt werden. Mal gilt es, in einem Wolkenkratzer eine Geiselnahme zu beenden, mal in einem abbruchreifen Stadtteil gemeingefährliche Hausbesetzer zu überwältigen. Wem das zu bodenständig ist, der kann mit einem "Gyropack" (der allerletzte Schrei im Robocop-Utilityshop) zwischen den Hochhäusern der Stadt herumdüsen. Der Spieler kann sich aussuchen, ob er die Geschichte des Films in einer Art 3D-Adventure nachspielen möchte oder lieber gleich in die Action-Einlagen, die er im Adventure erst nach und nach erreicht, einsteigen möchte. Das Spiel ist übrigens mit einem Dongle geschützt und enthält einen 16 x 17 cm großen Hologrammaufkleber. Abgesehen von diesen beiden Neuheiten, erwartet den Spieler ein robustes Action-Spektakel mit vielen fröhlichen Schußwechseln. ❑

(Heiner Stiller)



Peng, du bist tot – Verbrechensbekämpfung der Zukunft!

Robocop 3	
AMIGA DOS Testurteil 3'92	
befriedigend	60%
Grafik:	75%
Sound:	55%
Handling:	60%
Spielspaß:	60%
Preis/Leistung:	60%

Hersteller: Ocean
Muster von: Bomico
Preis: 89,95 DM
Konfiguration lt. Händler: A 500, A 1000, A 2000, 1 MByte

Hero Quest

Rollenspiele erfreuen sich allergrößter Beliebtheit, und das nicht nur als Computerversionen. Auch die brettgebundenen Versionen, die ja auch für alle Computer-Rollenspiele Pate standen, finden reißenden Absatz. Der einzige Punkt, der viele Spieler abschreckte, waren die enorm umfangreichen Regeln, die vor der ersten Partie durchgearbeitet werden mußten. An dieser Stelle setzte »Hero Quest« an, ein Brettspiel aus England, das Rollenspielvergnügen mit einem knapp gehaltenen Regelwerk verband. Das Computerspiel »Hero Quest« hält sich eng an die Brettspielvorlage. Bis zu vier Spieler können sich Helden aussuchen, die sie dann durch vorgegebene Szenarien lenken müssen, in denen es bestimmte Aufgaben zu erledigen gilt. Als Spielfiguren stehen Zauberer, Barbaren, Zwerge oder Elfen zur Auswahl. Alle besitzen besondere Eigenschaften, die sie von den anderen Figuren unterscheiden. Nacheinander kommen die Spieler an die Reihe und würfeln aus, wie weit ihr Spielcharakter sich bewegen darf. Pro Zug ist eine besondere Handlung wie die Durchsuchung des Raumes, Kampf oder Zauberei erlaubt. Auf diese Weise können die Spieler nun zusammen die gestellte Aufgabe erledigen oder aber auf eigene Faust handeln und sogar die anderen Mitspieler angreifen. »Hero Quest« ist ein sehr einfaches Rollenspiel, das alle Reize des Genres einzufangen vermag. Komplexe Handlungsverläufe wird man hier wohl vergebens suchen, Spaß machen die Abenteuer jedoch trotzdem. ❑

(hs)



In den Verliesen des Hexenlords ist die Hölle los

Hero Quest	
AMIGA DOS	
Testurteil 3'92	
gut	75%
Grafik:	75%
Sound:	70%
Handling:	70%
Spielspaß:	80%
Preis/Leistung:	75%

Hersteller: Gremlin
Muster von: Gremlin
Preis: 89,95 DM
Konfiguration lt. Händler: Amiga 500, 1000, 2000

Black Gold

Ungeachtet der Tatsache, daß Kohle ein Rohstoff ist, der subventioniert werden muß, daß es nur so kracht, haben die Programmierer von Starbyte eine Wirtschaftssimulation rund um das "Schwarze Gold" gezimmert. Sie versetzt den Benutzer zurück in jene Tage, in denen man sich mit der schwarzen Substanz noch eine goldene Nase verdienen konnte. Zentrales Spielthema sind, wie sollte es anders sein, der Kohlebergbau und der Verkauf damit geförderter Kohle. Mit einem eher bescheidenen Grundkapital ausgestattet, gilt es, ein Grubenfeld zu erwerben und alles zu besorgen, was zur Einrichtung eines funktionierenden Bergwerkbetriebes nötig ist. Vom Förderturm bis zum Holz, das für das Abstützen neu gegrabener Stollen gebraucht wird, reicht die Einkaufsliste des frisch gebackenen Bergbauunternehmers. Beginnt die Förderung des "Schwarzen Goldes", gilt es, Märkte zu erschließen und die Früchte der Spitzhacke gut zu verkaufen. Bis zu vier menschliche Spieler können an dem Spiel teilnehmen und wahlweise darum spekulieren, das meiste Geld auf das Konto zu bekommen, die meiste Kohle zu horten. Benutzerfreundlich ist »Black Gold«, daran gibt es nichts zu rütteln: Problemlos auf Festplatte installierbar, Sicherheitskopien sind möglich, und eine Speicheroption ist vorhanden. Insbesondere wenn mehrere Spieler an einer Partie teilnehmen, gewinnt das Spiel an Charakter und beginnt, Spaß zu machen. Ansonsten hat »Black Gold« eigentlich nur ein Problem: Gute Wirtschaftssimulationen gibt es schon reichlich, und allzuviel Variation läßt sich dem Thema nicht mehr abgewinnen. ❑

(hs)



Im Bergbau Kohle machen – eine neue Wirtschaftssimulation

Black Gold	
AMIGA DOS	
Testurteil 3'92	
befriedigend	60%
Grafik:	60%
Sound:	50%
Handling:	80%
Spielspaß:	60%
Preis/Leistung:	60%

Hersteller: Starbyte
Muster von: Compy-Shop
Preis: 89,95 DM
Konfiguration lt. Händler: Amiga 500, 1000, 2000; 1 MByte

Bunt gemischt

Neues bestimmt das Bild. In dieser neuen Rubrik werden wir versuchen, Sie über alle Neuheiten zu informieren, für die kein Platz ist, um sie in Form einer kompletten Review vorzustellen. Außerdem werden Sie auf den folgenden Seiten Interessantes über Budget-Spiele und Spielesammlungen finden. Wie Sie sehen, trägt die Rubrik den Titel "Bunt gemischt" nicht zu Unrecht.

Kid Gloves II

Kid Gloves Freundin wird gefangengenommen, und der Kleine Kid tauscht die Boxhandschuhe aus dem ersten Teil gegen einen Säbel, mit dem er sich zur Befreiungsaktion aufmacht. Mit »Kid Gloves II« bringt Millenium ein neues Jump-'n'-Run-Spiel der putzigen Sorte auf den Markt. Fünf Inseln, die sich jeweils aus mehreren verschiedenen Schauplät-



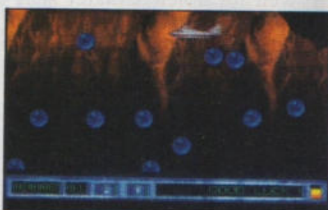
Kid Gloves II

zen zusammensetzen, muß Kid durchwandern. Bösewichter (kleine Bären, Pinguine, hüpfende Schneebälle und anderes Getier) kann sich Kid mit dem schon erwähnten Säbel vom Leib halten. Erlegte Gegner hinterlassen Geld, Schmuck oder die Schlüssel, die Kid braucht, um in das nächste Level zu gelangen.

Hersteller: Millenium
Preis: ca. 80 DM

Fantastic Voyage

Basierend auf einem bekannten utopischen Film aus den Sechzigern, entführt Sie dieses Spiel auf eine abenteuerliche Reise quer durch den menschlichen Körper. An Bord eines miniaturisierten U-Bootes müssen mutierte Körperzellen vor Ort bekämpft werden.


Fantastic Voyage

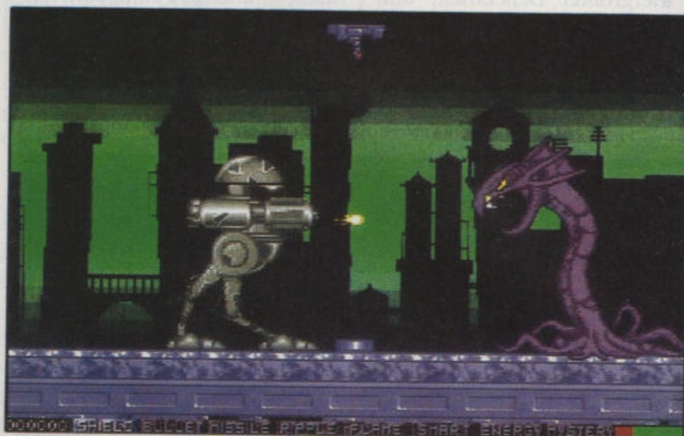
Diverse Extras sollen den Kampf gegen die Krankheitskeime erleichtern, müssen jedoch zunächst im Gewirr der Adern gefunden werden. »Fantastic Voyage« ist ein buntes Ballerspiel, das den Spieler stellenweise mit sehr origineller Grafik konfrontiert.

Hersteller: Centaur Software
Preis: 79,95 DM

Double Dragon III

Der Klassiker der Prügelspiele geht in die dritte Runde. Zwar geht es diesmal nicht um eine rüde Rachegeschichte, sondern um die Wiederbeschaffung gestohlener archäologischer Schätze. Daran, daß mit knakkenden Knöcheln gekämpft wird, hat sich jedoch nichts geändert. Zu der schon obligatorischen wüsten Prügelei haben sich jetzt noch Shops gesellt, in denen die Helden Schlagkraft, bessere Kampftricks und Extraleben kaufen können. Zwei-Spieler-Option bunte Grafik und derbes Spielgeschehen warten auf Amiga-Helden.

Hersteller: Storm
Preis: 79,95 DM



Under Pressure

Under Pressure

Als Kommandant eines klapprigen, bemannten Kampfläufers dürfen geneigte Amiganer besagtes Fortbewegungsmittel durch die Heimat einiger unfreundlicher Aliens steuern. Eine große Kanone dient der Hindernisbeseitigung, und aufgesammelte Muscheln bedeuten Extrawaffen, die für bunte Feuerwerke auf dem Bildschirm sorgen. Riesige Sprites und bildschirmgroße Monster, bei »Under Pressure« ist alles etwas größer als normalerweise. Das gemächliche Spieltempo macht »Under Pressure« auch für Ballerspielfreunde ohne lichtschnelle Reflexe interessant.

Hersteller: Electronic Zoo
Preis: 79,95 DM

Sliders

Ballspiel für zwei Kugeln und zwei Spieler. In diesem futuristisch angehauchten Spiel gilt es, mit einem Magnetball einen "Cyberpuck" (genau, wie das schwarze Ding beim Eishockey) im Tor des Gegners, das aus einer Pyramide mit ausgehöhlter Spitze besteht, unterzubringen. Beschleunigungsfelder und allerlei Bodenun-


Sliders

ebenheiten sorgen dafür, daß die gestellte Aufgabe nicht zu leicht fällt. Ein umfangreiches Optionsmenü läßt auch den blutigsten Anfänger die richtige Spieleinstellung finden, und



Double Dragon

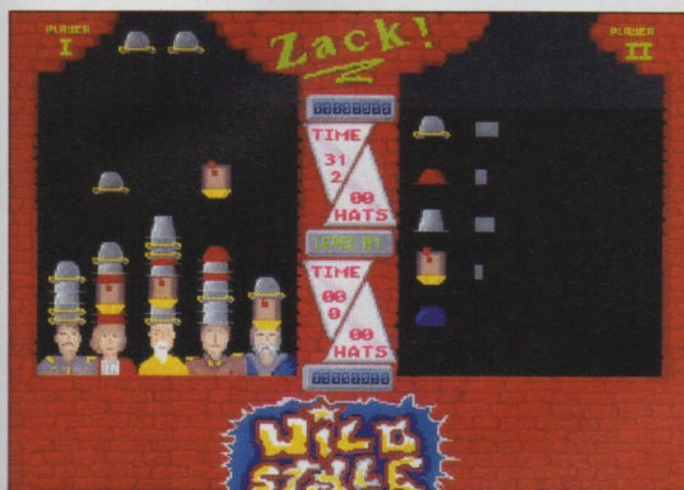
verschiedene Spielfeldlayouts lassen keine Langeweile aufkommen.

Hersteller: Palace
Preis: 79,95 DM

Zack

Hütestapeln, darin besteht die Aufgabe des Spielers bei »Zack«. Verschiedene Sorten von Kopfbedeckungen müssen jeweils in Dreiersortimenten zusammengebracht werden. Sind drei Kopfbedeckungen beisammen, verschwinden sie aus dem Spielfeld. Gelingt es nicht, diese Gruppen zu bilden, werden die Stapel immer höher, und wenn sie den oberen Rand des Bildschirms erreichen, ist das Spiel zu Ende.
Schnell erlernbares Spielprinzip und sanft ansteigender Schwierigkeitsgrad verbinden sich zu einem einfachen Geschicklichkeitsspiel, das solide Unterhaltung vermittelt.

Hersteller: Art Edition
Preis: 69,95 DM

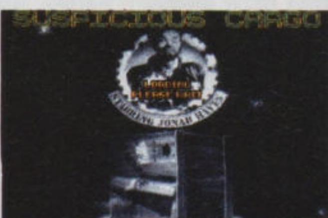


Zack

Suspicious Cargo

Abenteuerunterhaltung ganz neuer Art wird Amiga-Usern von diesem Programm versprochen. An Bord eines Transportraumschiffes, das zwischen Titan und Erde pendelt, gibt es Probleme mit der Ladung. Die ist offensichtlich lebendig und möchte nicht mehr in der Transportkiste bleiben.
Erstmals begegnen Sie hier auch Gremlins neuem Computerspielhelden, Jonah Hayes, von dem wir bestimmt noch Neues zu berichten haben werden.

Hersteller: Gremlin Graphics
Preis: 79,95 DM



Suspicious Cargo

The Dizzy Collection

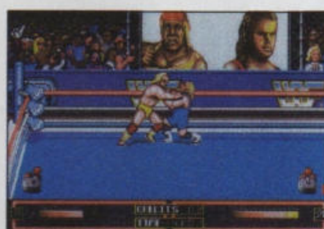
Das Ei Dizzy ist einer der beliebtesten Helden, die das Softwarehaus Code Master je kreiert hat. Nun gibt es die fünf Abenteuer des putzigen Patrons als Spielesammlung. »Fast Food«, »Kwik Snax«, »Treasure Island Dizzy«, »Fantasy World Dizzy« und »Magicland Dizzy« sind auf dem Sammler enthalten. Angesichts so vieler abwechslungsreicher Spiele sind Langeweileanfälle unwahrscheinlich.

Hersteller: Code Masters
Preis: 79,95 DM

4 Wheel Drive

Wer Rennspiele mag, der kommt um diese Sammlung nicht herum. Vier der besten Spiele dieses Genres, die das britische Softwarehaus Gremlin je herausbrachte, wurden hier zu einer attraktiven Sammlung zusammengestellt. »Lotus Esprit Turbo Challenge«, »Toyota Celica GT Rally«, »Team Suzuki« und »Combo Racer« geben sich ein PS-starkes Stelldichein.

Hersteller: Gremlin
Preis: 89,95 DM


WWF Wrestlemania

WWF Wrestlemania

Der beliebte Showsport Wrestling, hierzulande auch als Catchen bekannt, war immer wieder Thema von Computerspielen. Der neueste Vertreter dieser Sparte ist »WWF Wrestlemania«.
Unter den Ringern gibt es solche, die ihr Image eher als sympathische Kämpfer auslegen, und andere, die als Bösewichter durchgehen wollen. In diesem Spiel treten drei Ringer aus der Riege der "Guten" gegen drei aus den Reihen der "Bösen" an.
Der Spieler kann sich einen der 'Guten' aussuchen und mit diesem entweder Trainingsrunden absolvieren oder sich gleich auf den Kampf um den Meisterschaftsgürtel einlassen. Neben der Kampfsimulation liegt der Packung noch ein halbstündiges Video bei, auf dem man die Catcher in natura bewundern kann.

Hersteller: Ocean
Preis: 89,95 DM


The Dizzy Collection



4 Wheel Drive

Bart Simpson vs. the Space Mutants

Der kleine gelbe Giftzwerg ist inzwischen weltbekannt und neben seiner Karriere als Cartoonstar auch Held mehrerer Computerspiele.
In diesem Amiga-Programm darf er seine Heimatstadt vor der Invasion grauslicher außerirdischer Invasoren beschützen, die sich in den Köpfen seiner Nachbarn einnisten. Besessenen kann der gelbe Gnom auf den Kopf hüpfen und die Ärmsten dadurch aus den Klauen der Aliens befreien.
Nebenher gilt es noch, alle purpurfarbenen Gegenstände aus der Stadt verschwinden zu lassen, da die Außerirdischen das Zeug einsammeln und daraus die gefährlichste Waffe des Universums bauen wollen. Simpson-Fans werden sich freuen, zum Spiel gibt es noch einen kleinen Gummi-Bart zum An-den-Schlüsselbund-Hängen.

Hersteller: Ocean
Preis: 89,95


Bart Simpson vs. the Space Mutants

Antje Hink (hs)

Darkseed – Gigereske Visionen für den Amiga

Aliens im Computer

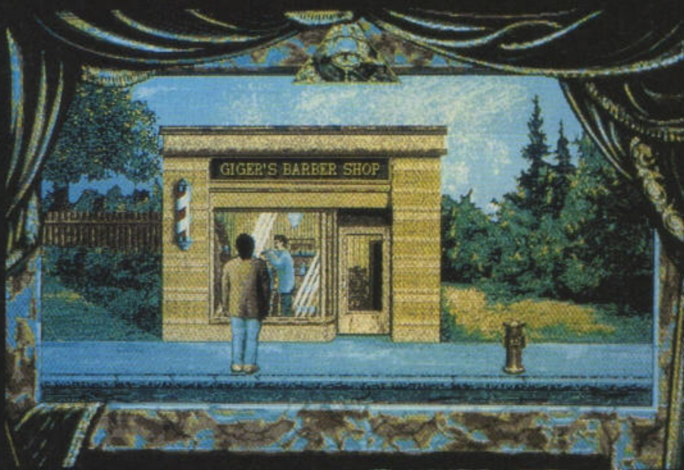
Wenn von Aliens die Rede ist, dann schießt jedem Zuhörer eine ganz bestimmte Form eines unheimlichen extraterrestrischen Wesens durch den Kopf. Der Film "Alien" machte vor gut zehn Jahren sowohl den englischen Begriff "Alien" (was unter anderem soviel wie "nicht wohlgesonnener Fremdling" bedeuten kann) rund um die Welt populär als auch einen ganz bestimmten damit assoziierten Grafikstil. Dieser Stil, der zur Darstellung des bösen Fremdlings wie auch dessen außerirdischer Umgebung verwendet wurde, ist der Phantasie des schweizerischen Künstlers Hans Rudi Giger entsprungen, der dafür auch einen Oscar, einen der begehrtesten Filmpreise der Welt, erhielt. Seine "biomechanoiden" Kreationen veränderten das Aussehen aller folgenden Filmungeheuer, und seine Nachahmer sind Legion.

Dem jungen Softwarehaus Cyberdreams gelang es, den inzwischen weltbekannten und vielbeschäftigten Künstler (zur Zeit arbeitet er an den Designs zum dritten Teil des Alien-Epos) zur Mitarbeit an einem Computerspiel zu gewinnen. »Darkseed« lebt von Gigers phantastischen Horrorvisionen einer Welt, in der Organismen und technische Komponenten zu einer beängstigenden Einheit fusionieren. Dementsprechend düster gestaltet sich auch die Handlung des Spiels, das als animiertes Grafik-Adventure angelegt ist und schon beinahe selbst filmmäßig abläuft: Ein junger Mann erlebt nach seinem Einzug in ein altes, geheimnisumwittertes Haus im Schlaf alptraumhafte Reisen in eine erschreckende Parallelwelt, in der er von einer Rasse böser Wesen erwartet wird. Diese hochtechnisierten Kreaturen pflanzen Dawson, so der Name des Protagonisten, einen Embryo in den Schädel, der nach drei Tagen schlüpfen wird. Als Dawson aus seinen Träumen erwacht, befindet sich das monströse Wesen tatsächlich in seinem Kopf und wird binnen drei Tagen schlüpfen und Dawson damit töten. Durch seine widernatürliche Geburt würde es außerdem für alle anderen Horrorwesen aus der Parallelwelt einen Zugang zu unserer normalen Dimension schaffen. Dawson muß also nicht nur sich selbst, sondern die ganze Welt retten.



Die »Darkseed« Dimension ist ein düsteres Abbild...

Im Verlauf des Spiels, das atemberaubende Grafiken beinhaltet und mit einer "Point-and-Click"-Steuerung ausgestattet ist, gelingt es Dawson, in seinem Haus einen Zugang zu der anderen Dimension zu finden. Dort muß er feststellen, daß alles, was hier in unserer "normalen" Realität existiert, auch eine Entsprechung in der gigeresken Parallelwelt besitzt. Jede Aktion im Hier und Jetzt beeinflußt auch die düstere Gegenwelt: Türen etwa, die in der Realität nicht geöffnet wurden, bleiben in der fremden Dimension verschlossen.





... unserer Realität

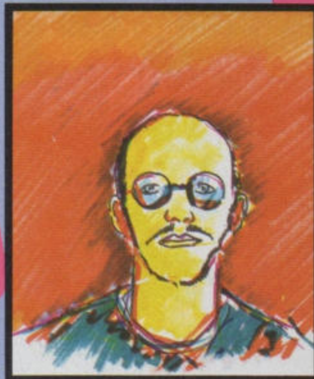
»Darkseed« befindet sich seit November 1990 in der Entwicklung und soll im November dieses Jahres auch für den Amiga erscheinen. Auch wenn es bis zur endgültigen Fertigstellung noch eine ganze Weile dauern wird, so ist »Darkseed« ein Programm, auf das es sich zu warten lohnt. ❑

(ab/bs)

Der heiße Draht

Fragen zu Amiga?

Fragen zur AMIGA DOS?



Thomas Baum
DFÜ, Grafik, Musik, Programmierung
Tel.: 05651 / 809- 7 40



Vera Brinkmann
PD und Software
Tel.: 05651 / 809- 7 43



Jürgen Borngießer
Hardware, BTX, Musik, Programmierung
Tel.: 05651 / 809- 7 42



Heinrich Stiller
Spiele
Tel.: 05651 / 809- 7 47



Garry Glendown
News, Tips & Tricks
Tel.: 05651 / 809- 7 41

Hotline: Di. & Do. 16.30- 18.00



Deluxe Paint IV

Erste Hilfe

»Deluxe Paint IV«, das unbekannte Wesen? Wir werden es Ihnen zeigen: In unserem dreiteiligen Schnellkurs begleiten wir Sie vom ersten Pixel über Brushes und Methoden zum Flächenfüllen bis zur kompletten Animation.

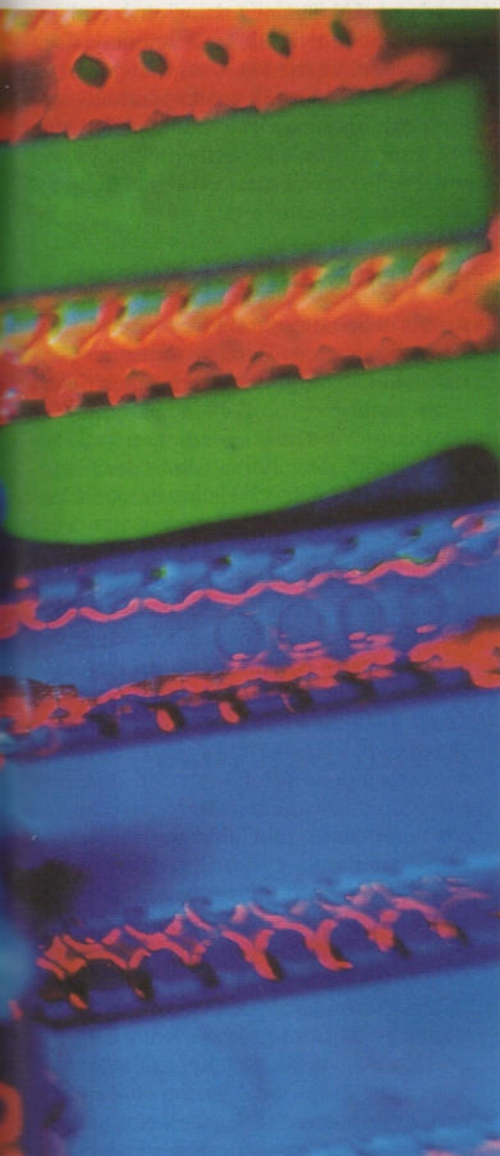
Kein Malprogramm ist wohl so verbreitet und bekannt wie »Deluxe Paint«. Den Erfolg bestätigt das Erscheinen der vierten Version, die einige beachtliche Funktionen bietet.
Doch umfangreiche Programme erfordern eine umfangreiche Bedienungsoberfläche. Obwohl das Handbuch sowie das Programm in deutscher Sprache verfaßt sind, fehlt vielen Einsteigern der Zusammenhang verschiedener Funktionen.
Wir wollen Ihnen den Einstieg in die Arbeit mit diesem flexiblen Malprogramm erleichtern und präsentieren Ihnen in einem dreiteiligen Einsteigerkurs die ersten Schritte zum Umgang mit diesem Grafikwerkzeug.

Existenzminimum und Luxus

Um mit »Deluxe Paint IV« arbeiten zu können, benötigen Sie einen Amiga mit mindestens 1 MByte RAM. Mehr Speicher ist besser, aber das haben Sie sich sicher selbst schon gedacht. Um das Programm komplett nutzen zu können, sollten schon mindestens 2 MByte verfügbar sein. Ein weiteres Kriterium ist die Größe des CHIPMEM. Der Amiga verwaltet zwei Arten von Speicher: CHIPMEM und FASTMEM. Im CHIPMEM müssen sämtliche Grafik- oder Sound-Sample-Daten abgelegt sein, wenn diese angezeigt oder abgespielt werden. Auch hier gilt: Je mehr CHIPMEM, desto besser. Ältere Amiga-Modelle verfügen meist nur über 512 KByte CHIPMEM. Besitzt Ihr Rechner bereits den neueren Agnus 8372, so können Sie mit einer Speichererweiterung 1 MByte CHIPMEM erreichen. Über diese Menge verfügen bereits die Amiga-Modelle 500 Plus und 3000, die dank des neuen Super-Agnus bis zu 2 MByte CHIPMEM verwalten können.
Als externer Massenspeicher reicht das interne Diskettenlaufwerk aus. Hier gilt das Gleiche wie beim Speicher: Mehr ist besser. Eine Festplatte ist empfehlenswert, ebenso wie eine Prozessorkarte, jedoch keinesfalls Bedingung.

Installation

Wenn Sie nur über ein oder mehrere Diskettenlaufwerke verfügen, können Sie direkt mit der Programmdiskette booten. Festplattenbesitzer haben noch etwas Arbeit vor sich, wenn »Deluxe Paint IV« auf der Harddisk installiert werden soll.
Richten Sie sich ein Verzeichnis auf der Festplatte ein, in das »Deluxe Paint IV« kopiert werden soll. Am einfachsten geht dies, indem Sie das Verzeichnis "Empty" kopieren und umbenennen. In den folgen-



den Schritten gehen wir davon aus, daß Ihr Verzeichnis "DH0:DPAINT" heißt.
Legen Sie nun die Programmdiskette in eines der Diskettenlaufwerke, und öffnen Sie die Disk mit einem Doppelklick. Sie finden ein Icon mit der Bezeichnung "Install DPaint". Klicken Sie dieses Icon zweimal an. Nach der Sicherheitsabfrage fordert das Installationsprogramm Sie auf, den Pfadnamen einzugeben, in dem das Programm installiert werden soll. In unserem Beispiel lautet er "DH0:DPAINT". Das Installationsprogramm kopiert nun alle nötigen Dateien. Dabei muß auch ein Zeichensatz (Font) in das Verzeichnis "FONTS:" kopiert werden. Dieses Verzeichnis ist das Fonts-Verzeichnis der Diskette oder der Festplatte, mit der gebootet wurde.
Hat das Installationsprogramm seine Arbeit beendet, klicken Sie das »Deluxe Paint IV«-Icon auf der Diskette an und ziehen es in den Deluxe-Paint-Ordner Ihrer Harddisk. Ebenso verfahren Sie mit dem Icon für "ColorText". Sollten Sie über OS 2.x verfügen, wird "ColorText" nicht benötigt.
Schließen Sie nun die Programmdiskette, und entfernen Sie die Disk. Sie legen die "ART1"-Diskette ein, öffnen sie und ziehen alle Icons in den Ordner Ihrer Harddisk. Ebenso verfahren Sie mit dem Inhalt der "ART2"-Diskette. »Deluxe Paint IV« ist jetzt installiert.

Es geht los!

»Deluxe Paint IV« wird durch Doppelklick auf das Programm-Icon gestartet. Sie haben auf dem nun erscheinenden Bildschirm die Möglichkeit, die Punktgröße Ihrer "Leinwand" sowie die maximale Anzahl der Farben einzustellen. Der Lo-Res- und Interlace-Bildschirm erlaubt die Benutzung von 32, 64 (Extra-Halfbright) oder 4096 (HAM) Farben. Bei Bildschirmen mit Med-Res- oder Hi-Res-Auflösung können nur höchstens 16 Farben eingesetzt werden. Verfügen Sie über wenig Speicher, kann sich die Anzahl der verwendbaren Farben unter Umständen verringern.
Der Overscan-Modus entfernt den Bildschirmrahmen. Die dadurch freiwerdende Fläche kann ebenfalls bemalt werden. Die Option "AUS" schaltet die normale Bildschirmdarstellung ein, "REG" und "MAX" vergößern den Screen entsprechend. Diese Optionen wirken sich nicht auf die Anzahl der Farben aus.
Die Einstellung "NTSC" oder "PAL" bestimmt den Videomodus des Amiga. NTSC ist die amerikanische Fernsehnorm. Sie bietet einige Zeilen weniger am unteren Bildschirmrand. In der europäischen PAL-Norm ist die ganze Bildschirmfläche für Ihre Maltätigkeiten frei.
Wenn Sie Speicherplatz sparen möchten, bietet sich die Option "Tauschen" an. Hier werden nur benötigte Programmteile im Speicher gehalten; es bleibt mehr RAM für Grafik frei. Hierbei werden jedoch später benötigte Programmteile nachgeladen und nicht benötigte aus dem Speicher entfernt. Diese Option sollten Sie einschalten, wenn Sie mit 1 MByte RAM arbeiten. Haben Sie mehr Speicher, können Sie "Alles laden" einstellen.
Für unsere ersten Experimente beschränken wir uns auf folgende Einstellungen: Lo-Res 320x256, 32 Farben, Overscan AUS, PAL. Sie klicken nun auf den OK-Schalter und befinden sich auf der Zeichenoberfläche.

Malen nach Zahlen?

Wenn Sie jetzt hauptsächlich Schwarz sehen, sehen Sie richtig. Schwarz ist die Hintergrundfarbe, die bei Programmstart eingestellt ist.
Im oberen Teil des Bildschirms befindet sich die Menüleiste, die zur Zeit lediglich den Kommentar "DeluxePaint Farbe" aufweist. Rechts oben befinden sich Gadgets, mit denen Sie »Deluxe Paint IV« in den Hintergrund (Workbench nach vorn) oder Vordergrund klicken können. Sollte dies im Moment keine Wirkung zeigen, so liegt dies daran, daß »Deluxe Paint IV« Speicher sparen will und die Workbench schließt. Wie Sie an Ihre Workbench wieder herankommen, verraten wir später. Aber seien Sie beruhigt: Spätestens nach Programmende taucht die Workbench wieder auf.
Das Wort "Gadget" könnte Ihnen noch unbekannt sein. Übersetzt bedeutet es nichts weiter als "Dingsbums". Offensichtlich ist den Entwicklern des Amiga kein besserer Begriff eingefallen. "Gadget" läßt sich am besten mit "Schalter" oder "Regler" umschreiben. Diese Gadgets lassen sich mit der Maus anklicken und führen dann irgendeine spezielle Programmoperation aus.
Auf der rechten Seite der Arbeitsfläche befindet sich eine Gadget-Leiste mit diversen Werkzeugen (im Englischen "Tools" genannt), mit denen wir uns nun beschäftigen werden. Um ein Symbol beziehungsweise eine Funktion auszuwählen, klicken Sie das betreffende Symbol einfach einmal mit dem Mauspfeil an. Hierbei gibt es Unterschiede zur rechten und linken Maustaste, die wir bei dem entsprechenden Symbol erklären werden.
Die obersten Symbole der Gadget-Leiste stellen Pinsel dar. Sie haben hier die Auswahl aus mehreren Pinselformen und Größen. Fahren Sie einmal mit dem Mauszeiger auf das große, kreisförmige Pinselsymbol in diesem Feld, und klicken Sie es einmal mit der linken Maustaste an. Jetzt plazieren Sie den Mauspfeil, der bereits den gewählten Pinsel mitführt, irgendwo auf der schwarzen Malfläche. Halten Sie die linke Maustaste gedrückt, und bewegen Sie die Maus. Sobald Sie die Taste loslassen, hört der Pinsel auf zu zeichnen. Sie sollten jetzt eine dicke Linie auf der Malfläche sehen.
Wählen Sie einen der anderen Pinsel an, und wiederholen Sie den Vorgang. Sie sehen nun deutlich die Unterschiede in der Liniendicke.
Das Wort "Brush", das immer wieder auftauchen wird, bedeutet nichts weiter als Pinsel. Pinsel können verschiedene Größen, Formen und Farben annehmen. Sogar mehrfarbige Brushes werden Sie kennenlernen.
Zunächst werden wir einen Pinsel in der Größe verändern. Klicken Sie mit der rech-

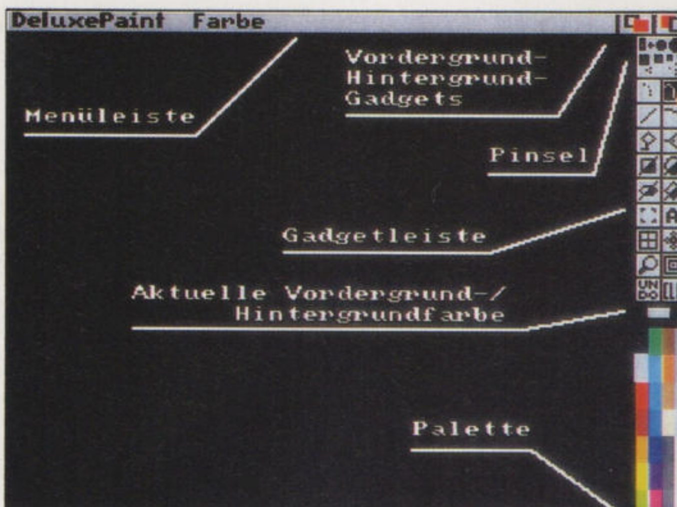


Bild 1. Die wichtigsten Bedienungselemente

ten Maustaste einmal auf den großen, viereckigen Pinsel. Bewegen Sie die Maus auf die Malfläche. Neben dem Pinsel ist nun das Wort "SIZE" erschienen. "SIZE" bedeutet Größe. Genau diese verändern Sie, wenn Sie die linke Maustaste gedrückt halten und die Maus nach unten rechts oder oben links bewegen. Wenn der Pinsel die gewünschte Größe erreicht hat, lassen Sie die rechte Taste einfach los. Sie können nun mit diesem Pinsel zeichnen.

Immer nur mit derselben Farbe zu malen wirkt etwas eintönig. Wir suchen uns also eine neue aus. Dazu dienen die bunten Quadrate am unteren Rand der Gadget-Leiste. Klicken Sie einmal eine Farbe Ihrer Wahl an. Über den Quadraten, die wir in Zukunft als "Palette" bezeichnen, wird ein Anzeigefeld entsprechend eingefärbt. Dieses Feld zeigt Ihnen ständig die aktuelle Zeichenfarbe an. Malen Sie nun mit der linken Maustaste ein wenig auf dem Bildschirm herum. Sie können dabei jederzeit aufhören und die Farbe wechseln.

Bisher haben wir zum Malen nur die linke Maustaste verwendet. Malen Sie einmal mit gedrückter rechter Taste über eine bereits bemalte Fläche. Die Farbe verschwindet, der Hintergrund wird sichtbar. Mit der rechten Taste wird also in der Hintergrundfarbe gemalt.

Wird Ihnen im wahrsten Sinn des Wortes der Bildschirm zu bunt, so können Sie ihn mit einem Mausklick auf das CLR-Symbol (unten rechts) löschen. Das Gadget mit der Aufschrift "Undo" macht die zuletzt ausgeführte Operation rückgängig. Klicken Sie einmal mit der linken Taste auf dieses Symbol: Das Bild ist wieder da. "Undo" funktioniert aber nur, wenn Sie nach der zu korrigierenden Aktion nichts anderes gemacht haben. Bedenken Sie: Immer die LETZTE Aktion wird rückgängig gemacht.

Die wenigsten Anwender von Zeichenprogrammen werden in der Lage sein, absolut runde Kreise oder gerade Linien aus der freien Hand zu malen. »Deluxe Paint IV« stellt uns eine Menge Funktionen zur Verfügung, um geometrische Figuren ohne ungewollte "Raumkrümmung" auf den Bildschirm zu zaubern. Diese Funktionen und einige mehr sind auf der Gadget-Leiste angeordnet, die wir jetzt einfach der Reihe nach ausprobieren.

Für unsere Experimente wählen Sie den kleinsten der verfügbaren Pinsel aus. Dieser Pinsel ist genau ein "Pixel" groß. "Pixel" bedeutet nichts weiter als Bildschirmpunkt. Ein Bildschirm von zum Beispiel 320 mal 256 Punkten ist also 81920 Pixel groß.

Direkt links unter der Pinsel-Auswahlbox befindet sich ein Gadget mit vier Punkten. Klicken Sie dieses Gadget einmal mit der linken Maustaste an, und malen Sie etwas auf dem Bildschirm herum. Bewegen Sie dabei die Maus unterschiedlich schnell. Sie werden bemerken, daß Sie nur dann eine einigermaßen durchgehende Linie zeichnen können, wenn Sie die Maus sehr langsam bewegen. Schnellere Bewegungen hinterlassen ein unregelmäßig punktierte Linie. Diese Funktion läßt sich wohl am besten mit "Freihandmalen mit Unterbrechungen" beschreiben.

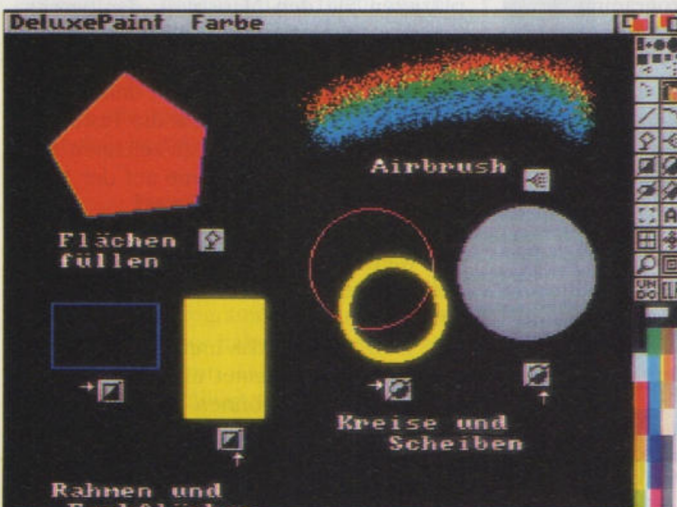


Bild 2. Geometrische Funktionen

Kreise, Linien, Kurven

Rechts neben diesem Gadget sehen Sie ein Gadget mit einer durchgezogenen "Kurve". Dieses Gadget ist durch einen Strich horizontal geteilt. Klicken Sie mit der Maustaste auf den oberen linken Teil dieses Gadgets, also oberhalb links der Diagonallinie, und malen Sie ein wenig. Sie bekommen eine durchgehende Linie, egal wie schnell Sie die Maus bewegen.

Etwas seltsamer sieht dieses Gadget aus, wenn Sie auf den unteren rechten Teil klicken, was Sie jetzt tun sollten. Malen Sie mit der Maus einen Umriß auf die Malfläche, dessen Anfangs- und Endpunkt auf demselben Punkt liegen. Sobald Sie die Maustaste loslassen, füllt »Deluxe Paint IV« die entstandene Fläche mit der aktuellen Farbe aus. Sie müssen übrigens den Anfangs- und Endpunkt nicht genau übereinanderlegen.

Das Gadget mit der diagonalen Linie, das sich unter dem "Punktierte Linie"-Gadget befindet, ist für das Linienziehen verantwortlich. Klicken Sie dieses Gadget einmal an, und bewegen Sie den Mauspfeil auf die Malfläche. Drücken Sie die linke Maustaste, halten Sie diese heruntergedrückt, und bewegen Sie den Mauspfeil an eine andere Bildschirmposition. Sie sehen eine gerade Linie, die den Bewegungen des Mauspfeils folgt. Wenn Sie nun die Maustaste loslassen, wird eine Linie vom Startpunkt zur aktuellen Pointer-Position gezogen. "Pointer" bedeutet "Zeiger". Damit ist, in unserem Fall, der Mauszeiger gemeint.

Bekanntlich gibt es nicht nur gerade Linien, sondern auch gebogene. Wenn diese gebogenen Linien auch noch regelmäßig sind, nennt man sie "Kurven". Solche Kurven können Sie mit dem Gadget rechts von dem Linien-Gadget erzeugen. Die Bedienung dieser Funktion ähnelt dem Linienzug: Startpunkt anklicken, mit gedrückter Maustaste den Endpunkt anvisieren und loslassen. Bewegen Sie anschließend den Mauszeiger auf der Malfläche herum. Sie sehen, daß die Form der Kurve der Zeigerbewegung folgt. Wenn Sie nun noch einmal mit der linken Maustaste klicken, wird die Kurve in der aktuellen Position gezeichnet. Übrigens: Diese Kurven werden Bezier-Kurven genannt (sprich: Besje-Kurven).

Unter dem Linien-Gadget befindet sich das Flächen-Füll-Gadget. Es stellt einen symbolischen Farbeimer dar, aus dem Farbe ausläuft. Zeichnen Sie mit Hilfe der Linien- oder Freihand-Funktion eine geschlossene Fläche. Wählen Sie nun eine andere Farbe, und klicken Sie das Füll-Gadget einmal "mit links" an. Der Mauszeiger verwandelt sich nun in eben jenen Farbeimer. Ganz unten in der Mitte des Pointers befindet sich eine kleine Lücke. Dies ist der "heiße Punkt" dieses Pointers, also das Pixel, mit dem gezielt wird. Bewegen Sie diesen Punkt in die zuvor umrandete Fläche, und klicken Sie dort hinein. Sofort wird die Fläche mit der gewählten Farbe gefüllt. Sollte sich unerwarteterweise der gesamte Bildschirm mit Farbe füllen, dann hatten Sie irgendwo ein Loch in der Umrandung der Fläche. In einem solchen Fall hilft die Undo-Funktion. Beim Aufspüren dieser Lücke kann Ihnen die Lupenfunktion helfen, die wir später noch eingehend erläutern werden.

Flächen füllen und Airbrush

Eine Art "Airbrush" stellt die Funktion rechts neben dem Füll-Gadget dar. Aktivieren Sie dieses Gadget, indem Sie einmal mit dem linken "Mousebutton" (so heißen die Tasten der Maus im "Profijargon") darauf klicken. Wenn Sie nun damit malen, wird eine Art Airbrush-Effekt erzeugt. Sie können für die Airbrush jede beliebige Farbe und jeden beliebigen Pinsel verwenden, auch mehrfarbige. Klicken Sie einmal mit der rechten Maustaste auf dieses Gadget. Sie können, wie schon von den Standard-Pinseln bekannt, die Größe des Airbrush-Bereichs verändern, indem Sie den Kreis auf dem Bildschirm entsprechend vergrößern oder verkleinern.
Die Funktion des Gadgets links unter dem Füll-Gadget haben Sie inzwischen sicher schon erahnt: Hiermit können Sie Boxen (viereckige Rahmen) und rechteckige Farbflächen erzeugen. Klicken Sie in die obere linke Hälfte des Gadgets, bewegen Sie den Mousepointer auf die Malfläche, und drücken Sie den linken Button. Sie haben in diesem Moment eine Ecke der Box festgelegt. Sie können nun durch Bewegen der Maus einen Rahmen auf dem Bildschirm zeichnen. Wenn der Rahmen die gewünschte Position hat, lassen Sie den Mausknopf einfach los.
Wenn Sie in den unteren rechten Bereich klicken, können Sie ebenfalls ein Viereck erzeugen, allerdings wird dieses gleich mit der aktuellen Farbe gefüllt.
Sehr ähnlich funktionieren die Kreis- und die Scheibenfunktion. Wenn Sie den oberen linken Bereich des Gadgets anklicken, können Sie einen Kreis zeichnen, der untere rechte Bereich liefert eine Scheibe (einen gefüllten Kreis). Die Vorgehensweise ähnelt den Boxen und Farbflächen, allerdings legen Sie zuerst den Mittelpunkt des Kreises fest und bestimmen (mit gedrücktem linken Mousebutton) den äußeren Rand des Kreises. Auch hierbei können jeder beliebige Pinsel und jede beliebige Farbe verwendet werden.

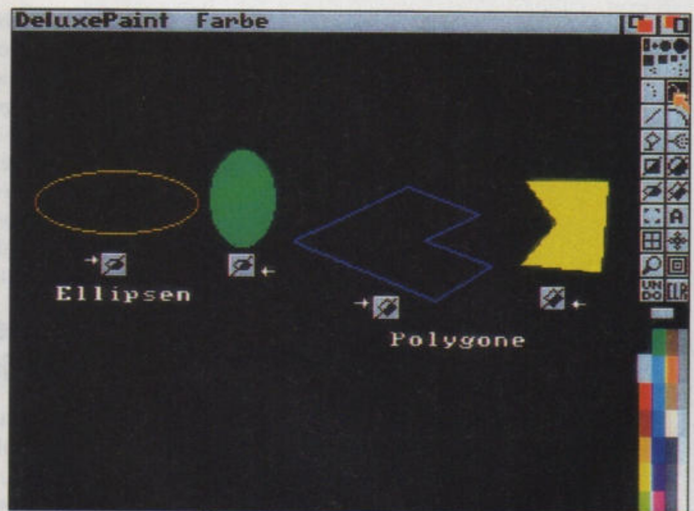


Bild 3. Unregelmäßige Geometrie

Eine Ellipse können Sie mit dem Gadget erstellen, das sich direkt unter dem Box-Gadget befindet. Klicken Sie dieses Gadget einmal in der oberen linken Hälfte an, und bestimmen Sie auf der Malfläche den Mittelpunkt der Ellipse. Klicken Sie nun einmal mit dem linken Mousebutton, und lassen Sie ihn los. Wenn Sie nun die Maus bewegen, ändert sich die Form der Ellipse. Eine horizontale Bewegung bestimmt die Breite der Ellipse, eine vertikale Bewegung legt die Höhe fest. Wenn die Ellipse Ihren Wünschen entspricht, klicken Sie einfach nochmals mit dem linken Button. Genauso verfahren Sie, wenn Sie eine ausgefüllte Ellipse malen wollen, nur daß Sie vorher den unteren rechten Bereich des Gadgets anklicken.

Rahmen, Kreise, Ellipsen und Polygone

Das Wort "Polygon" läßt sich am besten mit "Vieleck" übersetzen. Gemeint ist damit jede Fläche, die mehr als zwei Ecken besitzt. Dreiecke, Vierecke, Fünfecke ... alles sind Polygone. Ein Polygon braucht noch nicht einmal regelmäßig zu sein.
Solche Polygone können Sie mit dem Gadget neben dem Ellipsen-Gadget zeichnen. Auch hier – Sie haben es erraten – kommt es darauf an, welchen Bereich des Gadgets Sie anklicken. Oben links ist es ein ungefülltes Polygon, unten rechts wird die entstehende Fläche mit der aktuellen Farbe gefüllt. Ob gefüllt oder ungefüllt, die Vorgehensweise ist identisch. Klicken Sie einmal in den oberen linken Bereich des Gadgets und anschließend auf die Malfläche. Wenn Sie die Maus bewegen, werden Sie eine Linie sehen, die sich ähnlich wie die Linie des "Linienzugs" verhält. Klicken Sie nochmals mit der linken Maustaste, und bewegen Sie die Maus erneut. Sie sehen, daß eine neue Linie direkt an den Endpunkt der vorhergehenden angefügt wird. Zeichnen Sie einige Linien. Der Endpunkt der letzten Linie sollte auf dem Startpunkt der ersten Linie liegen. Fertig ist das Polygon. Experimentieren Sie ein wenig mit dieser Funktion herum; lassen Sie Linien sich überschneiden, und verwenden Sie auch den unteren rechten Teil dieses Gadgets.
Bevor wir uns im nächsten Teil auf Brushes (mehrfarbige Pinsel) und selbsterstellte Farbpaletten stürzen, möchte ich noch eine der Funktionstasten ansprechen.
Drücken Sie einmal die Taste [F10]. Die Menüleiste sowie die Gadget-Leiste sind verschwunden. Sie können den nun freien Bereich ebenfalls bemalen. Drücken Sie die Taste [F10] erneut, und die Menü-und Gadget-Leiste sind wieder da. ❑

Heike Albrecht (tb)

Kursfahrplan

Dieser Kurs wendet sich an alle Einsteiger in »Deluxe Paint IV«. Mit Einschränkungen sind die beschriebenen Funktionen auch für ältere Versionen dieses Programms verwendbar.
Teil 1:
Installation, Grundfunktionen und Beispiele
Teil 2:
Brushes, Texteingabe, Lupen-, Paletten- und Füllmusterfunktionen
Teil 3:
Metamorphose, Animationen, Tastaturkommandos und Tips

Musik-Werkstatt

Sounds à la carte

Es gibt sicher viele Gründe, einen Amiga zu kaufen. Für mich jedoch war einer besonders ausschlaggebend: Musik machen zu können, ohne ein Notenkünstler zu sein. Jedoch steht vor dem Preis erst der Fleiß; Musik will erst einmal entstehen, bevor sie gehört werden kann.

Da ich weiß, daß viele Einsteiger in diesem Gebiet Schwierigkeiten haben (schließlich habe ich ja auch einmal angefangen), will ich gerade denen, die gern selbst experimentieren möchten, in diesem Workshop Tips geben, wie sie zum ersten selbsterstellten Sound-Erlebnis gelangen – denn immer nur zuhören, was andere geschafft haben, frustriert auf die Dauer.

Von A wie "Aufnahme" bis Z wie "Zubehör"

Ziel unseres kleines Workshops soll das Erstellen von Samples sein, die wir in Musikstücke einbauen, die wir wiederum selbst erstellen. Dazu werden wir ein Programm benutzen, das mehrere Vorzüge in sich vereint:

– Es ist Sound-Editor für Samples,
– es enthält ein Sample-Programm,
– es ist in der Public Domain erhältlich und kostet nicht die Welt. Näheres dazu später.

Gemeint ist der »ProTracker«, einer von vielen Nachfolgern des berühmten »Sound-Tracker«, dem Urvater aller Sound-Editoren auf dem Amiga.

Bevor wir anfangen, Musik nach unserem Geschmack zu erstellen, werde ich Ihnen die "Werkzeuge" vorstellen, die ich in diesem Workshop benutze. Zuerst einmal die Voraussetzungen: Es sollte ein Stereo-Sampler vorhanden sein. Ich beziehe mich auf den in der letzten Ausgabe vorgestellten Techno-Sound-Sampler von Memphis. (Es ist allerdings fast jeder andere Stereo-Sampler ebenfalls tauglich, denn auf Spezifikationen der einzelnen Geräte wollen wir in diesem Workshop nicht eingehen.) Da Sie einen solchen Sampler im Besitz haben, setze ich voraus, daß Sie wissen, was Sie da gekauft haben. (Es sei denn, es regnet und der neue "Regenschirm" läßt sich partout nicht öffnen ...) Alle die, die planen, sich solch ein Gerät zuzulegen, sollten sich die kurze Erklärung innerhalb des Workshops durchlesen.

Als Tonquelle setze ich einen CD-Player voraus – auch der ist in jedem "gutsortierten" Haushalt inzwischen zu finden. Wer trotzdem keinen hat, kann auch ein Radio benutzen; es sollte allerdings über zwei sogenannte "Line-Out-Buchsen" verfügen. CD-Player geben ihr Signal direkt über einen Stereoausgang ab. Die dort anliegenden Signale können jedoch zu schwach sein. In diesem Fall lohnt sich noch der Einsatz eines Verstärkers. Bei den meisten Verstärkern ist neben den Lautsprecherausgängen ebenfalls noch ein Stereoausgang namens "Line Out" vorhanden, die dort anliegenden Signale wurden intern durch einen Signalverstärker aufgearbeitet. An den Lautsprecheranschlüssen hat der Sampler jedoch nichts verloren, Leistungsendstufen, die in Verstärkern für die notwendige "Power" sorgen, können den Digitalisierungs-Chip "erledigen". Im Grunde sieht der Anschluß aus, wie in Abb. 1 dargestellt. Als Verbindungskabel nehmen wir einpoliges, abgeschirmtes Audio-Kabel – sollte jedes gutsortierte Radio- oder Elektronikgeschäft in Kilometer-Ausführung vorrätig haben. Da nicht jedem das Löten liegt, kann man sich in solchen Geschäften auch Kabel herstellen lassen – es ist eine reine Geldfrage. Selbermachen ist billiger!

Die Stecker ergeben sich von selbst. Schauen wir die Audio-Seite des Samplers an, dann sehen wir zwei Cinch-Buchsen (manche besitzen dazu noch eine Mikrofonbuchse in 2,5/3,5 mm Ausführung, die soll uns jedoch jetzt nicht interessieren).

Ist alles so angeschlossen, wie in Abb. 1 zu sehen, kann es eigentlich losgehen. Wir schauen uns die Verbindung noch einmal an:

Audio-Tonquelle -> (Stereo-Verstärker mit "Line Out") -> Sampler-Audio-Eingänge -> Computer.

Noch ein Wort zu den Samples, die wir erstellen wollen: Da wir erst einmal üben müssen, sind Mitschnitte aus laufenden Radiosendungen oder CDs das beste Mittel, um mit dem Sample-Programm umzugehen. Instrumente mit Hilfe von Keyboards oder speziellen Sample-CDs zu erstellen ist etwas, was Sie nach diesem Workshop bestimmt mit Begeisterung noch machen werden – weil Sie dann auch das Wissen besitzen. Nur braucht man dazu ein gutes Gehör und ein wenig Übung. Aber jetzt wieder zurück zum Thema.

Der »ProTracker« ist, wie eben schon gesagt, ein Programm, das komplett alles enthält, was wir brauchen. Es hat allerdings ein paar Eigenheiten, auf die ich hier noch eingehen muß.

Zu bekommen ist der »ProTracker« zum Beispiel als Komplettpaket mit zehn Sample-Disketten mit zirka 400 Samples bei A.P.S. electronic (Adresse am Ende des Workshops). Wer keine Festplatte hat, sollte ihn auf eine formatierte Diskette kopieren und diese mit dem Namen "ST-00" benennen. Dies ist noch eine Eigenheit aus den Zeiten des »Sound-Tracker«. Der Name "ST-00" stand für die Programmdiskette, die Namen "ST-01" bis ST-XX" für die Sample-Disks. Dementsprechend sollte man eine ebenfalls formatierte Diskette mit dem Namen "ST-01" bereitlegen, sie soll unsere Samples speichern. Diese Namen sind unbedingt wichtig, denn »ProTracker« legt seine Konfigurationsdatei auf "ST-00". Zwar kann man in der Konfiguration den Such-

pfad nach eigenem Geschmack wählen, jedoch sollten wir alle Programmteile des »ProTracker« tatsächlich nach "ST-00" bringen, zumal (dazu kommen wir ebenfalls später noch) wir auch eine Sample-Liste brauchen – und auch die braucht "ST-00". Festplattenbesitzer sollten sich ein Verzeichnis namens "Sound" und innerhalb des Verzeichnisses die Unterverzeichnisse "ST-00" und "ST-01" anlegen. Diese Verzeichnisse müssen noch zu logischen Laufwerken verwandelt werden, dies geschieht durch den Befehl "ASSIGN" (näheres dazu in unserem CLI-Kurs).
Wenn wir diese Vorbereitungen erledigt haben, starten wir den »ProTracker«, indem wir die Workbench-Diskette starten, die »ProTracker«-Disk einlegen und den »ProTracker« mit Doppelklick laden.

"Es kratzt", oder "es ist dumpf"

Sehen wir uns den Haupt-Screen von »ProTracker« an, so finden wir rechts unten, über dem "Quadrascope", ein Feld mit Namen "Sampler". Wie wir schnell merken, ist der »ProTracker« weitgehend mausgesteuert, alle Felder lassen sich durch Mausklick aktivieren. Da uns im Augenblick nur der integrierte Sampler interessiert, klicken wir das Feld einfach an. Sofort ändert sich die untere Hälfte des »ProTracker«-Screens, und wir sehen ein freies Feld mit einer Mittellinie, auf der unser Sample grafisch dargestellt wird, sowie diverse Schalter, die ebenfalls per Mausklick eine Funktion aufrufen.
So, jetzt wird es spannend. Wir schalten unsere Audio-Tonquelle an und klicken das Feld "Sample" einmal unter Zuhilfenahme der linken Maustaste an. Der Mauszeiger färbt sich lila, und sofort müßten wir aus den Lautsprechern des Monitors, sofern wir die Tonausgänge des Amiga mit dem Monitor verbunden haben, genau das hören, was die Tonquelle, zum Beispiel der CD-Player, herausgibt. Außerdem sehen wir in dem Teil des Bildschirms, in dem vorher das vierteilige Quadrascope war, jetzt das anliegende Signal, und zwar grafisch dargestellt.
Wenn wir richtig hinhören, so merken wir schnell, daß die Wiedergabe vom Monitor "dumpfer" und "kratziger" klingt als das Original. Dies hat jedoch nichts mit der Qualität des Samplers zu tun, sondern entsteht durch den Rechner selbst. Der muß nämlich den Bildaufbau und das Einlesen des Signals gleichzeitig schaffen. Da Audiodaten und Bilddaten sich den Datenbus des Rechners teilen, geht die Aufnahmequalität erst einmal herunter. Drücken wir jetzt den Schalter "Sample", so endet die Wiedergabe. Damit haben wir aber noch keinen Sample. Wir haben aber erst das "Vorhören" kennengelernt. Hier kann man das anliegende Signal daraufhin beobachten, ob es vielleicht zu stark oder zu schwach hereinkommt. Ist es zu stark, würde der Sample Kratzgeräusche verursachen (man nennt dies "Oversampling"). Ist es zu schwach, würde der Sampler Störgeräusche besser digitalisieren als das eigentliche Tonsignal. Manche Sampler besitzen ein kleines Potentiometer (Einsteller) für einen bestimmten Arbeitsbereich, der zusammen mit einem richtig ausgepegelten Signal optimale Ergebnisse bringt. Besitzen Sie einen solchen Sampler (zum Beispiel der im letzten Heft vorgestellte »AURA1«-Sampler), sollten Sie eine Einstellung anhand des Handbuchs vornehmen.
Das Signal selbst läßt sich durch Lautstärkeregler abschwächen. Wir sollten daher den "Sample Monitor" beobachten (Drücken von "Sample" mit linker Maustaste) und nachschauen, ob das Signal dazu neigt, den oberen und unteren Rand dauerhaft zu durchbrechen – oder ob es ohne Lupe überhaupt zu sehen ist. Nehmen wir an, das Signal ist o.k., dann drücken wir jetzt (Achtung: Mauszeiger muß immer noch auf "Sample" stehen!) die rechte Maustaste.

Besser auf Disk als im Nirwana

Der Bildschirm wird eine Zeitlang dunkel, blendet wieder auf – und im Sample-Editor sehen wir unseren Sample Nummer eins. Jetzt gehen wir mit dem Mauszeiger auf das Feld "Waveform" und drücken die linke Taste. Es ertönt das gerade ausgeschnittene Teilstück. Drücken wir das Feld "Display", schallt es ebenfalls aus dem Lautsprecher, nur bei "Range" passiert nichts. Diese drei Felder dienen zum Abspielen des Samples, jedoch nur bei bestimmten Anforderungen. "Waveform" spielt alles, "Display nur das, was im Feld zu sehen ist (den Unterschied erkläre ich gleich), und "Range" nur Ausschnitte.
Rechts neben "Waveform" steht das Feld "Show Range", darunter "Show All". Klicken wir "Show Range" an, färbt sich der Mauszeiger kurz rot – und das war's. Wir haben also irgend etwas falsch gemacht. Das englische Wort "Range" heißt auf deutsch "Bereich". Da man Samples eigentlich nie so verwenden kann, wie man sie aufgenommen hat, ist es notwendig, bestimmte Bereiche zu isolieren. Beim »ProTracker«-Sampler geht das so: Gehen Sie mit dem Mauszeiger auf die Sample-Grafik, etwa bis zum Ende des ersten Drittels unseres Samples, drücken Sie die linke Maustaste, und halten Sie sie gedrückt. Ziehen Sie jetzt den Mauszeiger nach rechts bis etwa zum Ende des zweiten Drittels, und lassen Sie die Maustaste los. Der Bereich, den Sie mit der Maus abgefahren haben, wird jetzt invertiert dargestellt. Wir haben somit einen Bereich festgelegt. Drücken wir jetzt auf das Feld "Range", wird nur dieser Bereich gespielt. Wollen wir uns diesen Bereich genauer ansehen, betätigen wir das Feld "Show Range", schon wird der gewählte Bereich auf den Grafikeditor gebracht. Unter der Sample-Grafik sehen wir jetzt einen Balken, der

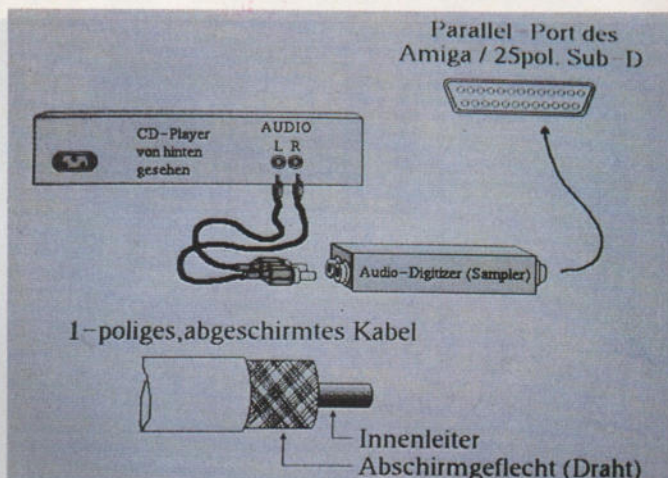


Bild 1. So sollte der Sampler an die Tonquelle angeschlossen werden

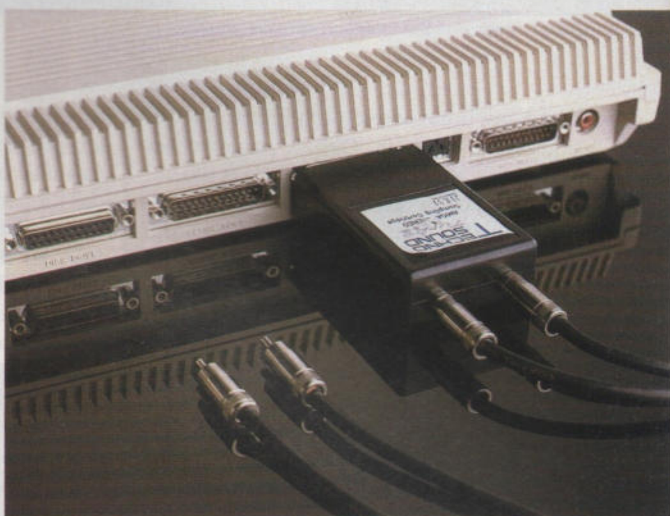


Bild 2. Sampler werden am Parallelport angeschlossen

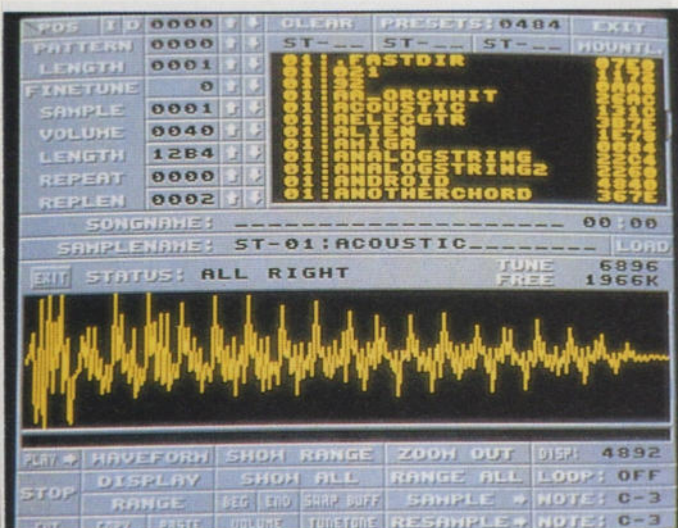


Bild 3. Der integrierte Sampler von »ProTracker«

den Ausschnittsbereich innerhalb der Gesamtlänge darstellt. Gehen wir mit der Maus auf diesen Balken und halten die linke Maustaste gedrückt, so können wir jeden Teil des Samples sehen, wobei die Ausschnittslänge gleichbleibt. Der Schalter "Display" spielt dabei nur den sichtbaren Teil des Samples ab.

So, kurze Verschnaufpause. Wir rekapitulieren noch einmal, was wir jetzt wissen: Wir haben die Tonquelle vorgehört und eingepegelt, einen Sample eingespielt, einen Ausschnitt markiert und wissen, auf welche Weise man Ausschnitte und den ganzen Sample abspielt.

Jetzt geht's weiter. Zuerst drücken wir auf "Show All", um des ganzen Samples Pracht wieder zu sehen. Der Sample befindet sich im Speicher, aber noch nicht auf Diskette. Ein Reset würde unsere Mühe einfach zunichte machen. Da hilft nur eins: abspeichern. Ich hoffe, Sie haben meinen Rat befolgt und eine Diskette namens "ST-01" formatiert oder ein logisches Laufwerk auf der Festplatte installiert, ansonsten: zurück zum Anfang. Alle anderen dürfen jetzt weitermachen.

Über dem Sampler steht ein großes Feld namens "Samplename". Dieses klicken wir an, daraufhin erscheint ein Eingabe-Cursor. Wir geben jetzt den Sample-Namen ein, nennen wir unser Meisterwerk einfach und schlicht "SMP1" und drücken [Return]. Eigentlich würde der Name für ein Abspeichern reichen, doch wir wollen unsere Samples auf "ST-01" unterbringen und nicht irgendwo. Also klicken wir das Eingabefeld noch einmal an, gehen mit dem Cursor nach links und geben ein "ST-01:", so daß jetzt komplett "ST-01:SMP1" dort steht. Sollten Sie die amerikanische Tastaturbelegung eingestellt haben, finden Sie den Doppelpunkt durch Drücken von [Shift]+[Ö], den Bindestrich durch Drücken der Taste [ß]. Jetzt gehen wir etwas höher, über dem "Quadrascope" finden wir "Disk Op.", gemeint ist "Disk-Operation". Klicken wir das Feld an, öffnet sich ein großes Fenster, zudem befinden sich jetzt Schalter für das Laden beziehungsweise Speichern von Samples, Songs und Modulen. Was Songs und Module sind, werden wir nachher kurz anreißen.

Jetzt drücken wir das Feld "Save Sample" an, darauf werden wir kurz gefragt, ob wir auch wirklich sicher sind, und nach Anklicken von "yes" (was sonst!) wird der Sample nach "ST-01" geschrieben – wobei "Disketten-Arbeiter" erst die richtige Disk einlegen müssen.

Unser erster Sample befindet sich jetzt komplett auf der Sample-Disk – und deshalb können wir jetzt fröhlich an ihm "herumschnippeln". Dazu begeben wir uns wieder in den Sample-Editor.

Das Markieren eines bestimmten Bereiches haben wir ja schon vorgenommen, genau dasselbe machen wir jetzt mit unserem Sample noch einmal: Mauszeiger auf das Ende des ersten Drittels, linke Maustaste drücken und festhalten, nach rechts ziehen bis zum Ende des zweiten Drittels und Maustaste loslassen. Jetzt dürfte, wie vorhin, das mittlere Drittel des Samples wieder invertiert erscheinen. Schauen wir uns jetzt die drei Felder unter "Range" und "Stop" (dient zum Beenden des Abspielens) an; sie nennen sich "Cut", "Copy" und "Paste". Nehmen wir zuerst "Cut" – schneiden. Klicken wir dieses Feld an, wird der markierte Bereich rigoros herausgeschnitten. Laden Sie den Sample wieder (einfach "Disk Op." und "Load Sample" anklicken, danach wieder in den Sampler gehen), und probieren Sie "Copy" und "Paste" aus. Ersteres kopiert den ausgewählten Bereich in einen *Puffer-Speicher*, das zweite setzt diesen Puffer an die Stelle des Samples, wo der Cursor steht.

Schauen wir uns das bildlich an: Nehmen wir an, unser Sample ist ein Satz mit dem Wortlaut "Billy-the-Bit". Wir markieren den Bereich "the-", wählen das Feld "Copy" an. Danach steht "the-" im Puffer. Jetzt wählen wir "Cut", und als Sample haben wir nur noch "Billy-Bit". Jetzt gehen wir mit dem Cursor an den Anfang des Samples und drücken kurz auf "Paste". Der Inhalt des Puffers, also "the-" wird in den Sample zurückgeschrieben – und schon steht da "the-Billy-Bit". Ist doch einfach, oder?

Der Ton macht die Musik – der Editor

Schauen wir uns noch die fünf Felder unter "Show All" an. Hier finden wir zum Beispiel die Cursor-Markierungen "Beg" (Begin = Anfang) und "End" (Ende), die einfach nur der Markierung dienen, sowie "Swap Buff". Haben wir einen Bereich herauskopiert, so befindet der sich im Puffer. Mit "Swap Buff" schalten Sie einfach zwischen Sample-Speicher und Puffer um. "Volume" erlaubt das Einstellen der Lautstärke des gesamten Samples oder eines Bereiches. Die Ausgangslautstärke und die Endlautstärke können per "Schieberegler" und Maus verändert werden, die Skala wird dabei in Prozent angezeigt. Interessant ist, daß die Option "Ramp" nutzbar ist. "Ramp" läßt einen Ton abschwellen, wenn "\" (down) angeklickt wurde, und ansteigen, wenn "/" (up) angeklickt wurde. "Tunetone" gibt einen Grundton aus, genauer gesagt den Ton "A" mit einer Frequenz von 440 Hertz. Diesen Ton kann man zum Angleichen des Samples an die Tonleiter benutzen, die den Sample mit den Feldern "Sample" und "Note" sowie "Resample" und "Note" abspielt oder neu erstellt (dazu mehr in einem späteren Workshop). Wer will, kann ja schon einmal probieren. "Range All" markiert den gesamten Bereich, "Zoom Out"

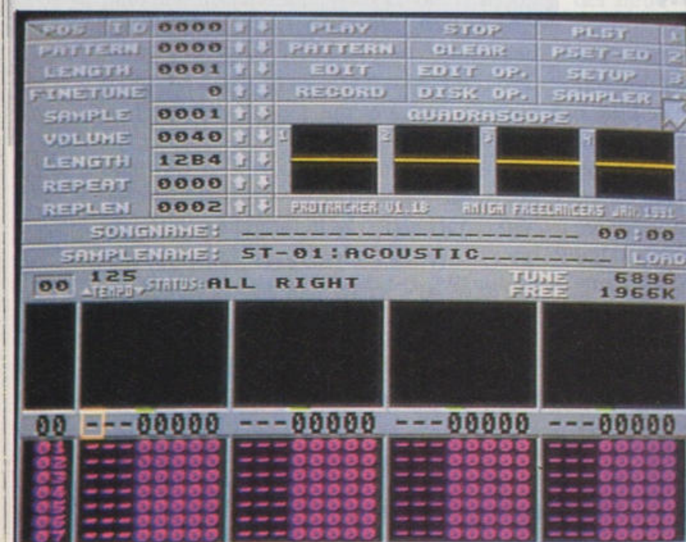


Bild 4. »ProTracker« mit den vier Eingabefeldern für die Kanäle

holt den markierten Bereich "näher heran". "Loop" ermöglicht ein dauerhaftes Wiederholen des Abspielens, "Disp" zeigt die Länge des Samples im Display in Bytes an.
Das war also der Sampler innerhalb des »ProTracker«, gleich kommen wir zum Editor.
Doch erst noch ein Päuschen mit gleichzeitiger Rekapitulation: Wir wissen jetzt, wie man Teile aus Samples ausschneidet, in einen Puffer kopiert und von dort wieder einsetzt. Und wir können Samples leiser und lauter werden lassen.

Schneiden, kopieren, einsetzen

Was wir bis jetzt geschafft haben, ist noch nicht das, was wir erwarteten. Der Grund hierfür ist einleuchtend: Mit einem einzigen Sample, der zudem noch aus einem zufällig herausgeschnittenen Teil stammt, ist noch keiner in die Hitparade gekommen – auch wenn es sich teilweise so anhört. Hier, genau hier, wenn Sie diesen Text jetzt lesen, sollten Sie erst einmal abbrechen und üben, Samples erstellen, wieder löschen, "schneiden", "kopieren", "kleben" – und vor allem üben. Man braucht dafür ein gutes Gehör, vor allem Gefühl für Takt. (Es reicht nicht, Frau oder Freundin die Tür aufzuhalten – mit Takt ist der Rhythmus gemeint!) Und so sollte die Hauptaufgabe sein, ein tatsächliches Instrument zu kreieren, das sich tatsächlich für Musikstücke eignet. Möglich wäre dazu beispielsweise der Einsatz von speziellen Sample-CDs, die Instrumente in teilweise exzellenter Form wiedergeben. Der Nachteil ist jedoch der Preis; ich selbst habe mir einmal eine CD mit Synthesizer-Sounds geleistet und mußte stolze 70 DM dafür über die Ladentheke schieben. Außerdem kann man noch so aufpassen, das Musikstück, dessen Rhythmus sich hervorragend zum "Sämmpeln" eignet, kommt garantiert immer dann im Radio, wenn man den Amiga gerade in mindestens zehn Kilometer Entfernung stehen hat. Aber, wie gesagt, manchmal hat man vielleicht doch Glück, und dann sollte man schon wissen, was man zu tun hat. Also, üben – und zwar jetzt ...
... Sind Sie wieder da? Na prima, dann wollen wir uns noch den eigentlichen Editor vom »ProTracker« ansehen. Ich will hier keine Einführung in die Befehle des »ProTracker« geben, denn erstens wird das Programm mit einer Dokumentation ausgegeben, und außerdem ist diese Dokumentation als "Online-Hilfe" jederzeit über die Taste [Help] aufrufbar.
Wichtiger erscheint mir, Ihnen kurz zu zeigen, wie bestimmte Samples im »ProTracker« eingesetzt werden können. Dazu laden wir noch einmal den "Ur-Sample", den wir erstellt haben (Sie wissen doch hoffentlich noch, wie?), und klicken das Feld "Exit" des Samplers an. Es erscheint wieder das Bild wie zu Anfang unseres Workshops: In vier Spalten befinden sich Zahlenfelder, die zu Anfang mit lauter Nullen gefüllt sind. Drücken Sie jetzt die Taste [Cursor down] (Pfeiltaste nach unten). Die Zahlenreihen wandern nach oben, die linke Zahlenreihe zählt aufwärts bis 63. In der Mitte der Zahlenfelder befindet sich das Editorfeld, in dem die Eingaben gemacht werden können. Was wir hier sehen, ist ein sogenannter "Pattern", ein Stück aus einem Song, den wir noch nicht erstellt haben. Ein Pattern besteht aus 64 Takten (0-63), die sich in der Geschwindigkeit variieren lassen; dazu mehr im Workshop der nächsten Ausgabe. Wichtig ist erst folgendes: Wenn wir unseren Sample geladen haben, dann steht dieser als "Sample 0001" im Speicher – zu sehen am gleichnamigen Feld. Theoretisch können wir jetzt schon editieren. Versuchen wir es einmal.
In der oberen Hälfte des Editors finden wir das Feld "Edit". Dieses klicken wir an, im gleichen Augenblick färbt sich der Mauspfeil blau, das heißt, daß eine Eingabe erwartet wird. Mit den Cursor-Tasten [auf] und [ab] gehen wir jetzt auf "Takt 0" (nennen wir es erst mal so). Wenn wir jetzt eine Taste drücken ([Y] bis [-], [S], [D], [G], [H], [J], [L], [Ö], [Q] - [U], [2], [3], [5], [6], [7]), erhalten wir unseren Sample in einem der ausgewählten Grundtöne. Schauen wir uns die belegten Tasten näher an, fällt uns auf, daß sie in etwa einer Klaviatur gleichen. Dementsprechend schnell ist es möglich, Musik zu erstellen, ohne Noten lesen oder schreiben zu müssen. Wenn wir auf diese Art eine Note eingegeben haben, nehmen wir an, durch Druck auf Taste [Y], dann sollte jetzt im Editorfeld für Kanal 1 die Buchstaben-Zahlen-Folge "C-201000" stehen. C-2 steht für den Ton C in Oktave 2, 01 bedeutet "Sample 01", insgesamt 31 Samples sind möglich (hexadezimal benannt von "01" bis "1F"). Der Rest der Zahlenkolonne soll Thema des nächsten Workshops sein. Ich will Ihnen nur ein kurzes Beispiel geben, wie Sie Ihren Sample so einsetzen können, daß sich ein sinnvoller Takt ergibt. Der »ProTracker« teilt eigentlich eine Note in 64 Einzelschritte auf (vorsichtig ausgedrückt). Je nach Abspielgeschwindigkeit kann man nun im 2/4-Takt, im 4/4-Takt, im 4/16-Takt arbeiten, je nach Aufteilung. Im Grunde muß der eigene Takt nur durch vier teilbar sein. Teilt man die 64 Schritte durch "4", erhält man 16. Das heißt, ein 4/4-Takt müßte bei den Schritten "16", "32" und "48" neu beginnen. Versuchen wir auch das. Wir gehen wieder in den "Edit-Modus" und drücken die Taste [Y] an den Positionen 0, 16, 32 und 48. Drücken wir jetzt auf "Pattern" (Feld über "Edit"), hören wir unseren Sample im 4/4-Rhythmus. Übrigens

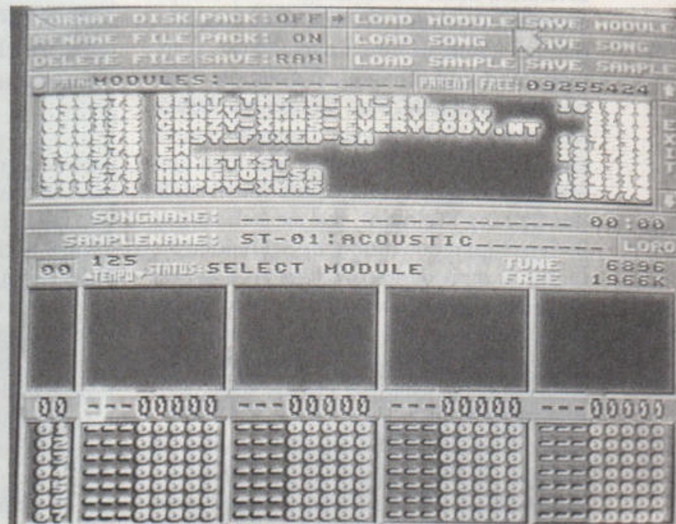


Bild 5. Fertige Songs werden platzsparend als Module abgespeichert

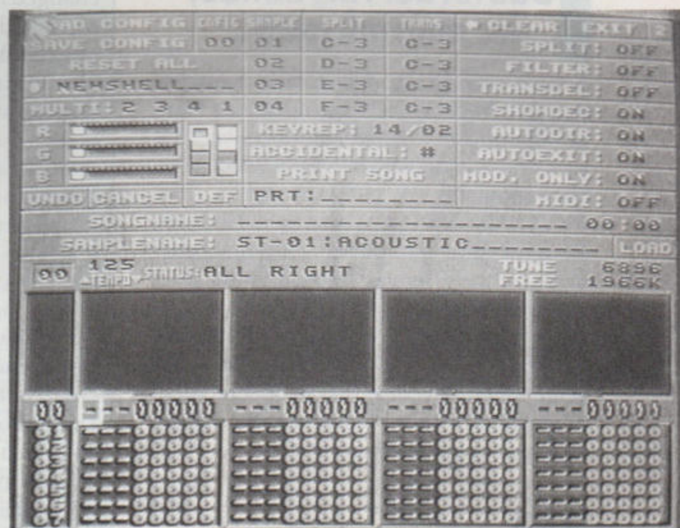


Bild 6. Das Setup-Menü 1 des »ProTracker«

kann man sich den Akt mit den Cursor-Tasten sparen. Drücken Sie einfach nacheinander [F5], [F6], [F7] und [F8], und das Edit-Fenster wandert automatisch 16 Schritte weiter.
Den Sound, den wir da erstellt haben, kann man wohl getrost als "Sondermüll" bezeichnen, denn weder Länge noch Tempo stimmen. Doch das dürfte ja wohl klar sein, daß ich kaum zu Ihnen nach Hause kommen kann und Ihnen auch noch die Samples fertig erstelle. Statt dessen mache ich jetzt erst mal Schluß, gebe Ihnen aber auch eine Aufgabe mit: Erzeugen Sie doch einmal einen Sample aus einem Musikstück der Gattung "Tekkno", wie sie sich zuhauf in den Hitparaden tummeln. Schneiden Sie diesen Sample dann so, daß er als 4/4-Takt einen flüssigen Übergang hat. Beim nächsten Mal wollen wir uns nämlich anschauen, wie man eine Rhythmusbegleitung erstellt, und da sollte so etwas "schon aus dem FF (hex) klappen".
Auf unsere DATABOX zum Heft speise ich ein paar Probe-Samples ein, die mit dem »AudioMaster III«, einem Profi-Sampler der Spitzenklasse, erstellt wurden. Wer Interesse hat, kann ja mal reinhören.

Das ist der Rhythmus, wo man mit muß ...

Ach ja, wer gleich richtig viele Samples, den »ProTracker« und dazu noch jede Menge Supermusik will, der kann sich das "Music Construction Set" bei A.P.S. electronic, Sonnenborstel 31, 3071 Steimbke, Tel. 05026/1700, für 50 DM besorgen. Über 400 Samples sind auf dieser Zusammenstellung, teilweise etwas seltsam, dafür aber gut zum Probieren.
Beim nächsten Mal erkläre ich Ihnen die Funktionen des »ProTracker« genauer und zeige Ihnen Beispiele für besondere Techniken bei der Sound-Erstellung. Bis dahin – viel Spaß beim "Sämmpeln". ❑

(jb)

Warum Musik digitalisieren?

Die Antwort auf diese Frage müßte eigentlich lauten: weil's Spaß macht! Tatsächlich gibt es heute kaum noch Musik in der Pop-Szene, in der nicht ein findiger "Sound-Kopierer" seinen Digitizer im Spiel hatte. Mit digitalisierten Sounds läßt sich so manches Programm mit dem berühmten "Pep" versehen. Aber Vorsicht! Alle Samples müssen praktisch als Raubkopien gelten, wenn das Urheber-Recht nicht gewahrt ist. Deshalb immer eine Quellenangabe machen!

Sampler – was ist das?

Manche reden vom Sampler und meinen ein Programm, andere reden vom Sampler und meinen einen kleinen, meist schwarzen Kasten, der auf der einen Seite nach Computer aussieht, auf der anderen dafür eher wie ein Zubehörteil für die heimische Stereoanlage.

Beide Gruppen liegen richtig – und doch wieder falsch. Doch woher soll man als Anfänger wissen, was gemeint ist?
Kurz gesagt, der Sampler besteht aus beiden Komponenten, Hard- und Software. Wenden wir uns zuerst einmal der Hardware zu. Sie ist das Wichtigste, wenn man digitalisieren will. Und damit haben wir auch den eigentlichen Namen gefunden: Audio-Digitalisierer. Von Video-Digitizern hat man im Bereich Amiga bestimmt schon gehört, sie wandeln Video-Bilder von Kameras oder Recordern beziehungsweise Playern in ein Format, das der Amiga verwalten kann, zum Beispiel HAM (Hold and Modify) oder IFF (Interchanged File Format).
Nicht anders kann man mit Audio-Signalen verfahren. Was man braucht, ist ein kleiner integrierter Elektronikbaustein, der die Audio-Signale in digitale Signale umwandelt. Da man in der Regel kein komplettes Musikstück digitalisiert (es sei denn, man ist einer der wenigen Glücklichen, die 16 MByte Speicher besitzen – ich beneide sie), nimmt man einen Ausschnitt oder ein Stück vom Ganzen. Und da Englisch im Computerbereich "in" ist (man kann Tante Frieda immer so schön beeindrucken), nennt man das Stück vom Ganzen "Sample". Aha!
Da Audio-Signale analog sind, das heißt zwischen ganzen Werten auch jeden Zwischenwert annehmen können, ist es notwendig, diesen Digitalisierer einzusetzen. Digital heißt ja, daß nur zwei Zustände ("An" oder "Aus" – "1" oder "0") möglich sind. Nehmen wir an, wir müßten die Zahl "5" digital darstellen. Da wir hier das Zweier- oder Dualsystem anstelle des Dezimalsystems benötigen, müssen wir die Zahl aufsplitten. Die Zahl "2" geht insgesamt zweimal in die Summe "5", der Rest ist "1". Rechnen wir nach:
2*2+1=5
dual also
2^2 + 2^0 (da in die Eins keine Zwei paßt)
= 2^2=4 + 2^0=1 = 5
Kein Problem also beim Digitalisieren.
Was ist aber, wenn wir die Realzahl 5,2535 digitalisieren wollen? Im Vertrauen gesagt, auch das kann man dual darstellen und berechnen, aber wir wollen uns lieber ansehen, was der Sampler daraus macht: Ein kompliziertes Netz aus Widerständen innerhalb der integrierten Schaltung sorgt für eine Aufteilung der ganzen und der Bruchzahlen, die in einem bestimmten Verhältnis zueinander in Digits (Dualzahlen) umgewandelt werden. Audio-Daten werden so Wert für Wert umgewandelt und eingelesen.
Gerade weil der Digitizer in der Lage ist, auch Zwischenwerte zu digitalisieren, haben wir die Möglichkeit, Bilder und Töne in den Rechner zu bringen. Und hier kommen wir zu einem großen Manko der "normalen" Sampler: Da sie an den Parallelport angeschlossen werden, kann die digitalisierte Zahl nur einen acht Bit breiten Datenkanal benutzen. Bei ganzen Zahlen bedeutet dies einen Wertebereich von "0" bis "255". Da Audio-Daten grundsätzlich aus vielen Zwischenwerten bestehen, jedoch nur die Acht-Bit-Möglichkeit besteht, geht die Qualität der digitalisierten Daten deutlich "in die Knie". Wer viel Geld ausgeben will, kann sich inzwischen 12- oder 16-Bit-Sampler besorgen, die eine enorme Qualitätssteigerung zulassen, gleichzeitig aber immensen "Speicherverbrauch" haben. Aber, wie gesagt, Geld sollte man haben – zirka 1000 DM für eine 12-Bit-Karte, zirka 2200 DM für eine 16-Bit-Karte muß man schon hinblättern.
Doch zurück zu unseren Samplern. Das Digitalisieren ist klar, nur müssen wir die Bits und Bytes auch in den Speicher bekommen. Dazu gibt es Sample-Programme, die die ankommenden Daten nacheinander ins RAM schreiben, zudem noch verschiedene Möglichkeiten besitzen, das einmal digitalisierte Stück Musik zu verändern, zu verbessern – oder auch zu verschlechtern, wenn man nicht aufpaßt.
Ein Sampler ist also: Tonquelle -> Digitalisieren -> Einlesen -> Verändern – wie man sieht, mehr als nur zwei dahingeworfene Begriffe.



Einsteiger-Kurs

Programmieren mit ARexx

Seitdem Kickstart 2.0 und mit ihm der »ARexx«-Interpreter ausgeliefert wird, stellt sich Amiga-Deutschland eine Frage: Was kann ich mit »ARexx« anfangen? Eine interessante Frage, die im folgenden beantwortet werden soll.

Liest man alle Veröffentlichungen, die zum Thema »ARexx« erschienen sind und sicher auch noch erscheinen werden, bleibt die Qual der Wahl, welches denn nun die richtige Definition des vielgepriesenen "Wunderwerks" »ARexx« ist. Die ernstzunehmenden Beschreibungen schwanken in der Regel zwischen "einer BASIC-ähnlichen Programmiersprache", "einer Standard-Script-Sprache" und "einem Tool zur Kommunikation zwischen verschiedenen Programmen". Um es vorwegzunehmen: Alle haben irgendwie recht.

Der Kern des Pakets besteht aus einem sogenannten Interpreter, der den Sprachumfang des Systems beinhaltet. Ein Interpreter ist ein Programm, welches eine Computersprache (hier speziell die Sprache »Rexx«) in eine für den Prozessor verständliche Form, die Maschinensprache, übersetzt.

Im Gegensatz zum Compiler, der dies pro Programm einmal macht und das Ergebnis als startbare Binärdatei abspeichert, übersetzt ein Interpreter das Programm, während es läuft. Dieses Vorgehen hat sowohl Vor- als auch Nachteile, was hier aber nicht näher betrachtet werden soll (lesen Sie dazu auch unser Titelthema in dieser Ausgabe). Es soll nur festgehalten werden, daß »Rexx« (beziehungsweise die Amiga-Version »ARexx«) das Programm (den Sourcecode) interpretiert.

Gestartet wird der Interpreter durch Aufruf des Programms »rexxmast«, welches sich im Verzeichnis »System« der Workbench-Diskette befindet. Dieses Programm installiert einen Hintergrundprozeß, der darauf wartet, daß ein »ARexx«-Programm mit der Bitte um Ausführung an ihn herangetragen wird. Als Bittsteller fungiert das Shell-Kommando »rx« im Verzeichnis »Rexxc« der Workbench.

»ARexx« – Ersatz für BASIC?

Beim Start werden ihm als Parameter der Name des »ARexx«-Programms sowie eventuelle Parameter für das Script übergeben, welches dann an den Hintergrundprozeß weitergereicht wird. Ermöglicht wird diese Vorgehensweise durch die Multitasking-Fähigkeiten des Amiga-Betriebssystems und das Message-System der AmigaOS-Schaltzentrale »Exec«. Alle Programme, die durch »rx« gestartet werden, erwartet der Interpreter im logischen Gerät »Rexx:«, Standardsuffix (Endung der Programmnamen) ist ".rexx". Diese Vorgaben werden somit als bekannt vorausgesetzt und können weggelassen werden. Als erstes Programm kommen Sie natürlich nicht um den außerordentlich einfallsreichen Namen "Hello World" herum, geben Sie also mit einem beliebigen Texteditor das Script

Hinweis: Falls Sie weiter mit »ARexx« programmieren wollen, dann schauen Sie auch in die "Tips & Tricks", dort ist ein Artikel über die Steuerung vom »PowerPacker« durch »ARexx«.

```
/* Erstes »ARexx«-Script */
say "Hello World"
```

ein, speichern Sie es unter dem Namen »Rexx:Hello.rexx« ab, und starten Sie es anschließend mit

```
rx hello
```

»Say« entspricht dem Befehl »Print« in BASIC, dem »WriteLn« von Pascal oder dem »Echo« der Amiga-Shell und gibt eine Zeichenkette aus. Die beiden Zeichenfolgen "/*" und "*/" kennzeichnen einen Kommentar, daß heißt, alles was zwischen ihnen steht, wird vom Interpreter ignoriert und hat somit keinen Einfluß auf den Programmablauf. Jedes »ARexx«-Script muß mit einem beliebigen Kommentar beginnen.

Die Sprache an sich, entwickelt Mitte der achtziger Jahre vom Amerikaner M. F. Cowlishaw, ähnelt in gewisser Weise wirklich dem bekannten BASIC, auch wenn diese Ähnlichkeit nicht so weit geht, wie einige Leute dies anscheinend gerne hätten. Neben der leichten Erlernbarkeit, die beiden Sprachen eigen ist, sind die größten Gemeinsamkeiten mit Sicherheit in den wirklich komfortablen Bearbeitungsmöglichkeiten von Zeichenketten (Strings) zu finden. »Rexx« geht hier sogar noch einen Schritt weiter als BASIC, der Vorrat an String-Funktionen ist schier unerschöpflich. So können Zeichenketten beliebiger Länge durch einfaches "Hintereinanderschreiben" zusammengefügt werden. Die Zeile

```
neuerString = alterString" nochwas"
```

verbindet den Inhalt der Variablen »alterString« mit der Zeichenkette »nochwas«. Ist »alterString« nicht als Variable definiert, was durch eine einfache Zuweisung geschieht, ist Name gleich Inhalt. Ist aber vorher eine Zuweisung an »alterString« erfolgt, beispielsweise

```
alterString = "Dies und"
```

wird der Inhalt der Variablen verwendet, in diesem Beispiel wäre das Ergebnis also "Dies und nochwas". Die Teile einer Zuweisung, die immer direkt in das Ergebnis übernommen werden, sind also in Anführungszeichen einzupacken. Sollen zwei Variablen direkt miteinander verknüpft werden, müssen deren Namen mit einer leeren Zeichenkette voneinander getrennt werden, also

```
neuerString = alterstring1 ""alterString2
```

Einfacher gehts eigentlich schon nicht mehr.

Aber auch komplexere String-Funktionen sind zur Genüge vorhanden. Einen detaillierten Überblick über die vorhandenen Funktionen und die Kontrollstrukturen (Schleifenkonstrukte ...) verschafft Ihnen das leider nur eventuell vorhandene Handbuch (Commodore hat es leider versäumt, dem A500 Plus ein »ARexx«-Handbuch beizulegen) oder der Workshop ab AMIGA DOS 8/91. Hier geht es mehr um das Gesamtkonzept der Sprache und im speziellen um die Integration der Amiga-Implementation in eine bestehende Arbeitsumgebung.

»ARexx« und die Variablen

Ein weiteres Hauptmerkmal der Sprache »Rexx« sind die typenlosen Variablen. Andere bekannte Hochsprachen wie C oder Oberon verlangen vom Programmierer eine sogenannte Variablendaklaration. Dies ist eine Aufstellung, die dem Compiler mitteilt, welche Variablen der Programmierer zu benutzen gedenkt, und vor allem, was jede einzelne Variable beinhalten soll. Unterschieden wird hierbei beispielsweise zwischen Zahlen, einzelnen Zeichen und Zeichenketten. Man spricht dabei von verschiedenen Variablentypen. »Rexx« kennt diese Unterscheidungen nicht, alle Variablen sind typenlos oder, das ist reine Definitionssache, vom gleichen Typ. Der Inhalt jeder Variablen kann jeder anderen ohne irgendwelche Beschränkungen zugewiesen werden. Für jede Funktion, die eine solche Variable als Parameter übergeben bekommt, bedeutet dies, daß sie zuerst prüfen muß, ob sie mit den ihr übergebenen Werten wirklich etwas anfangen kann. Eine mathematische Funktion beispielsweise wird sich erst einmal vergewissern, daß sie im folgenden auch mit Zahlen und nicht etwa mit irgendwelchen Zeichenketten arbeiten wird.

Was auf den ersten Blick wie ein Nachteil ausieht (immerhin werden die Programme dadurch langsamer), entpuppt sich beim zweiten Hinsehen als ein Vorteil, wenn man sich die umgekehrte Richtung der Zuweisung (eine reine Zahl an einen String) ansieht. Strings stellen eine Art Obermenge der Variablentypen dar, sprich: Ein String kann alle bekannten Zeichen in beliebiger Reihenfolge beinhalten. Das Ergebnis einer Berechnung kann also später ohne Probleme mit jeder String-Funktion bearbeitet werden, um beispielsweise für die Ausgabe in einer Tabelle formatiert zu werden. Da »ARexx« nie als Programmiersprache für Anwendungen gedacht war, nicht zuletzt, weil Interpreter-Sprachen naturgemäß ohnehin nie die schnellsten sind, überwiegen hier die Vorteile die Nachteile bei weitem.

Willkommen in "Port ARexx"

Erweitert werden kann der Grundbefehlssatz von »ARexx« durch externe Funktionsbibliotheken (function libraries). Sie stellen dem »ARexx«-Programmierer Funktionen zur Verfügung, die er in eigenen Programmen verwenden kann. Dieses Konzept entspricht in etwa dem der Shared-Libraries, die Sie für gewöhnlich in »Libs:« vorfinden.

Die zweite, wesentlich mächtigere Erweiterungsmöglichkeit stellen die bekannten »ARexx«-Ports dar. Eine Anwendung, die über einen solchen Port verfügt, stellt dem Hintergrundprozeß beziehungsweise dem laufenden »ARexx«-Programm eine Art Briefkasten zur Verfügung, in welchem dieser beliebige Nachrichten hinterlassen kann. Diese Nachrichten beinhalten Befehle, die das Anwendungsprogramm versteht und anschließend bearbeiten kann. Eventuelle Ergebnisse (Erfolg, Mißerfolg, Benutzereingaben während der Befehlsausführung ...) werden dem »ARexx«-Script wieder zur Verfügung gestellt. Jede Applikation, die einen »ARexx«-Port besitzt, stellt erst einmal einen gewissen Grundstock an Kommandos zur Verfügung, die im "Amiga User Interface Style Guide" beschrieben

ARexx-Tips gesucht!
»ARexx« ist die Sprache , die den Amigas der neueren Generation beiliegt. Leider wissen die wenigsten Einsteiger, welche Möglichkeiten der Programmierung ihnen »ARexx« bietet – wohl auch deshalb, weil Besitzer eines A 500 PLUS ohne Handbuch auskommen müssen.
Aus diesem Grund gibt es diesen Kurs – und aus diesem Grund suchen wir Tips rund um »ARexx«. Wenn Sie ein interessantes ARexx-Script geschrieben haben, senden Sie es an die Redaktion AMIGA DOS. Bei Veröffentlichung winkt als Lohn ein großzügiges Honorar.

Tabelle 1

Name	Parameter	Funktion
New	Portname/k Name des zugehörigen neuen Ports, der wirklich vergebene Name wird zurückgegeben	Neues Projekt starten, inklusive neuer Arbeitsumgebung
Clear	force/s wenn gesetzt, wird der User nicht darauf aufmerksam gemacht, daß er noch nicht gespeichert hat	Arbeitsumgebung löschen
Open	Filename/k Name der Datei force/s siehe Clear	Neue Datei in alte Arbeitsumgebung
Save		Datei speichern. Wenn namenlos, erscheint ein Filerequester
Saveas	Filename/k gewünschter Name	Unter dem gegebenen Namen speichern
Close	force/s siehe Clear	Projekt schließen und Arbeitsumgebung verlassen
Print	prompt/s wenn gesetzt, Requester für die Druckereinstellung öffnen	Dokument ausdrucken
Quit	force/s siehe Clear	Programm verlassen
Cut		markierten Block ausschneiden und am Clipboard zwischenspeichern
Copy		Kopie des markierten Blocks am Clipboard zwischenspeichern
Paste		vom Clipboard ins Dokument einfügen
Erase	force/s wenn nicht gesetzt, Sicherheitsabfrage	markierten Block löschen
Help	Command/s Befehl, der erläutert werden soll prompt/s aktiviert ein intuitiongesteuertes Hilfssystem.	Hilfstexte anzeigen
Fault	Fehlernummer/n	Schreibt eine Fehlerbeschreibung in die ARexx-Standard-Variable Result
Rx	Console/s gesetzt, wenn ein Con-Fenster zur Ausgabe benötigt wird Async/s wenn gesetzt, läuft das Script asyncron zum Aufrufer Command/f wird so an den Hintergrundprozess weitergegeben	Ein ARexx-Script starten

Tabelle 1. Die vom "Style Guide" vorgegebenen Kommandos

und allgemeingültig sind, sprich sie müssen in jedem Programm auf die eine oder andere Art vorkommen. Dies sind Befehle zum Öffnen und Schließen von Dateien, zum Kopieren oder Löschen von Datenblöcken (in Malprogrammen zum Beispiel entspricht dies dem Ausschneiden von Pinseln, in einem Texteditor dem Kopieren oder Verschieben von Textabschnitten) etc. Alle vom "Style Guide" vorgegebenen Kommandos inklusive einer kurzen Beschreibung befinden sich in Tabelle 1.

Jeder Programmierer, der sein Programm nach diesen Richtlinien gestaltet und die erwähnten Kommandos implementiert, erleichtert dem Anwender den Einstieg in sein Produkt und erspart ihm unnötige Umgewöhnungszeit. Befehle, die in jeder Anwendung gleich sind, merkt man sich einfacher und schneller als irgendwelche exotischen Bezeichnungen, die im Prinzip auch nichts anderes machen. Weiterhin beschreibt der "Style Guide" eine Menge Kommandos, die nicht in jedem Anwendungsprogramm Sinn machen, aber falls sie es doch tun, ebenfalls den Richtlinien folgend implementiert werden sollten.

Hinzu kommen eine unbestimmte Anzahl an anwendungsspezifischen Kommandos. So wird eine Textverarbeitung zum Beispiel Funktionen bereitstellen, die den Text an der Cursorposition einliest, wohingegen bei einer Datenbank im Datenbestand gesucht werden kann, vollautomatisch wohlgemerkt. Im Grunde kann man sagen, daß ein guter »ARexx«-Port alle Funktionen, die das Programm in Pull-down-Menüs oder Gadgets anbietet, kennt.

Wohin mit den Befehlen?

Angesprochen wird eine solcher Port mit dem »ARexx«-Befehl »address«, dem der Name des Ports (der wird von der Anwendung festgelegt) sowie ein eventueller Befehl folgt. Fehlt der Befehl am Ende, bedeutet dies, daß der angegebene Port nun der aktuelle ist und alle folgenden Kommandos an diesen einen Port zu schicken sind, solange es sich nicht um »ARexx«-interne Befehle handelt. Ansonsten wird nur dieser eine Befehl an den angegebenen Port geschickt und danach wieder der vorherige aktuelle gewählt. So erklärt das Script

```
/* Address-Beispiel eins */
address command
address powerpacker ezrequest "Beispiel"
type "s:startup-sequence"
```

die Shell (Portname command) zum aktuellen Port, wendet sich dann mit dem Befehl »ezrequest« an den »PowerPacker«, um anschließend die »Startup-Sequence« auszugeben.

Die Variante

```
/* Address-Beispiel zwei */
address command
address powerpacker
ezrequest "Beispiel"
type "s:startup-sequence"
```

hingegen funktioniert nicht wie gewünscht, weil »Type« ein Shell-Kommando ist und somit nicht vom »PowerPacker«, dessen Port ja als letzter zum aktuellen erklärt wurde, ausgeführt werden kann. Die Zeile "address command" bewirkt somit rein gar nichts, ihr Ergebnis wird vom Nachfolger "address powerpacker" regelrecht vernichtet. Der aktuelle Port wird immer dann angesprochen, wenn der auszuführende Befehl weder dem »ARexx«-Sprachumfang zugeordnet werden kann, noch in einer der vorhandenen externen

Funktionsbibliotheken zu finden ist. Weiß auch das Anwendungsprogramm, dem der angesprochene Port gehört, nichts mit dem Kommando anzufangen, weist es dieses mit einer entsprechenden Fehlermeldung zurück, und die Abarbeitung des »ARexx«-Scripts wird abgebrochen. Für Sie als Anwender haben all diese Fähigkeiten den Vorteil, daß Sie häufig wiederkehrende Arbeiten im Programm automatisieren beziehnungsweise den Befehlsumfang sogar bis zu einem bestimmten Grad eigenhändig und auf Ihre Bedürfnisse hin zugeschnitten erweitern können. Vor »ARexx« gab es bereits einige Programme, die diese Möglichkeiten mit Hilfe einer eigenen Script-Sprache boten. Gegenüber dem »ARexx«-Konzept hat dieses aber einige gravierende Nachteile. Erst einmal ist es für den Programmierer wesentlich weniger aufwendig, einen »ARexx«-Port anstatt einer kompletten Script-Sprache zu implementieren, was wohl bedeutet, da Sie insbesondere bei kostengünstigen Programmen viel häufiger in den Genuß dieser Möglichkeiten kommen werden. Hinzu kommt, daß die Script-Sprache aller Programme die gleiche ist, nämlich »Rexx«. Sie als Anwender müssen nicht für jedes Programm eine andere, wenngleich meist ähnliche Sprache lernen, Sie können auch bei einem brandneuen Produkt auf Ihr altes Wissen zurückgreifen. Lediglich die neuen anwendungsspezifischen Kommandos müssen in der Dokumentation nachgeschlagen werden. Ein Umgewöhnen entfällt völlig.

Ein Beispiel zum Üben

Der dritte und mit Sicherheit wichtigste Vorteil entspringt der Möglichkeit eines jeden »ARexx«-Programms, mehrere Ports verschiedener Applikationen nacheinander ansprechen zu können. Auch hier wirken sich die Multitasking-Fähigkeiten des AmigaOS entscheidend aus. Dies hat zur Folge, daß sich mehrere gleichzeitig im Speicher stehende Programme gegenseitig fernsteuern und synchronisieren können, um dann zum Schluß Daten untereinander auszutauschen. Im praktischen Einsatz könnte so beispielsweise ein Script entstehen, das einem Texteditor mit Hilfe einer Datenbank zu einer Serienbrieffunktion verhilft, die, je nachdem, ob der einzelne Empfänger über eine Netzwerkadresse verfügt oder nicht, den Brief entweder mit der richtigen Adresse versieht und ausdruckt oder mit Hilfe eines Point-Programms, welches ebenfalls einen »ARexx«-Port bereitstellen muß, den Brief über das Netz verschickt. Der grobe Ablauf dazu findet sich in Abbildung 1.

Die Funktionsweise des Scripts in Abbildung 1 ist recht einfach nachzuvollziehen. Es sollte vom Editor gestartet werden, wobei der Brief bereits geladen sein muß. Anschließend wird in der Datenbank automatisch die Adreßkartei geöffnet, eine Suchbedingung, nach der die Empfänger des Briefs festgelegt werden, abgefragt (zum Beispiel alle "Kleinpopelsdorfer Adressen") und der nächste gefundene Datensatz, der diese Bedingung erfüllt, in einer temporären Datei zwischengelagert. Zusätzlich wird die eventuell vorhandene Netzwerkadresse in einer Variablen gespeichert. Nun tritt wieder der Texteditor in Aktion, der die temporäre Datei (also die kurz vorher gefundene Adresse) einliest und im Serienbrief plaziert. Falls der Empfänger nicht über das Netzwerk erreichbar sein sollte, wird der Brief ausgedruckt und mit dem nächsten Adressaten fortgefahren. Ist er es, wird das Point-Programm beauftragt, den Brief als elektronische Nachricht zu versenden, da dies in der Regel schneller und kostengünstiger ist. Der Vorgang wiederholt sich so lange, bis keine Adresse mehr vorhanden ist. Die ganze Sache läßt sich natürlich noch beliebig verfeinern, indem beispielsweise nur bestimmte Datenfelder eines jeden Datensatzes in den Brief übernommen werden oder indem gefragt wird, ob ein Brief trotz vorhandener Netzwerkadresse ausgedruckt und per "Sackpost" verschickt werden soll.

So weit der Ausblick auf die Möglichkeiten, die »ARexx« Ihnen als Anwender bietet. Voraussetzung ist natürlich eine entsprechende Unterstützung von Seiten der Programmierer, aber die Veröffentlichungen der letzten Zeit haben gezeigt, daß eigentlich kaum ein Programm, das etwas auf sich hält, ohne »ARexx«-Port auskommt. Nur eines sollten Sie sich merken: »ARexx« ist kein Ersatz für eine der herkömmlichen Hochsprachen wie C oder Modula beziehungsweise Oberon und wird auch nie einer werden. Vielmehr ist die Sprache ein Teil des Benutzerschnittstellen-Dreigestirns, zu dem zusätzlich noch Intuition, die grafische Oberfläche und die Shell gehören. ❑

Christoph Teuber (jb)

Literaturverweis
Addison Wesley (1991): Amiga User Interface Style Guide, ISBN 0-201-57757-7

```
(* Start Kasten 1 *)

/*
   Hierbei handelt es sich nicht um ein lauffähiges ARexx-Programm
   sondern mehr um einen umgangssprachlich formulierten groben
   Ablaufplan der das System ansich gut verdeutlicht. Die Voraus-
   setzungen sind ein Texteditor, eine Datenbank und ein Pointprogramm,
   welches den Versand der elektronischen Nachricht (per Modem)
   übernimmt.
   Alle durchgehend groß geschriebenen Worte mit deutschsprachigen
   Bezeichnungen stellen fiktive Kommandos dar, die vom jeweils
   aktuellen Port bereitgestellt werden. Die englischsprachige
   Bezeichnungen finden Sie in Tabelle 1 wieder, sprich es handelt sich
   um Standard-ARexx-Kommandos, die jede Applikation besitzen sollte.
   Alles in eckigen Klammern beschreibt die Funktion, die ein hier
   einzusetzender Platzhalter einnehmen soll. "Netzwerkadresse" ohne
   eckige Klammern ist eine normale Variable, alles andere sind
   reguläre Rexx-Befehle.
*/

address Datenbank
OPEN <Adresskartei>
ABFRAGE <Suchbedingungen für Serienbriefempfänger>

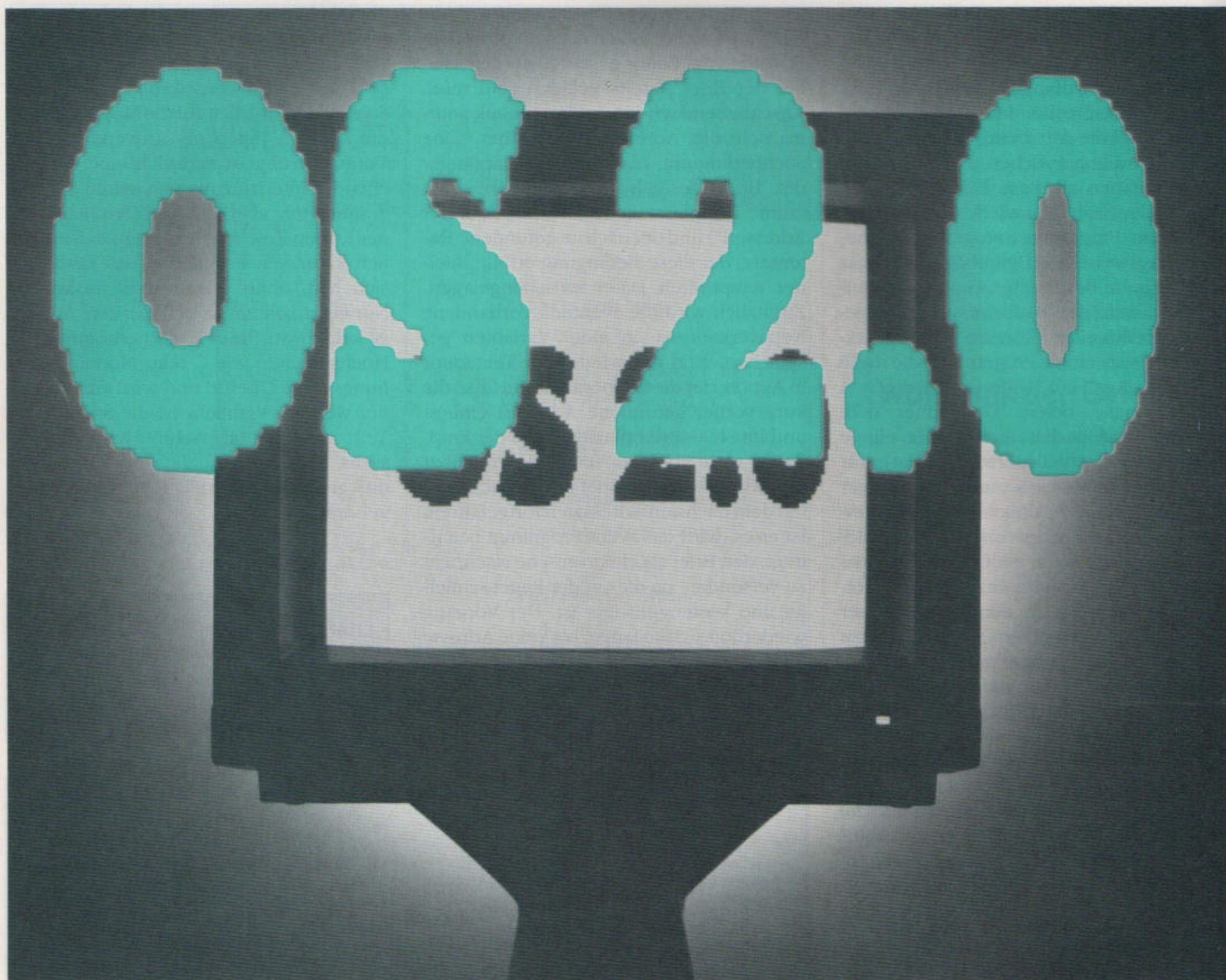
Repeat
 address Datenbank
 SUCHE <nächster Datensatz der der Suchbedingung entspricht>
 EXPORTIERE <gefundenen Datensatz> <in temporäre Datei>
 Netzwerkadresse = HOLE_FELD_AUS_DATENSATZ <Netzwerkadresse>

 address Texteditor
 FÜGE_EIN <temporäre Datei> <in Brief>
 If Netzwerkadresse = "" then     /* leer, gleich keine Netzwerkadresse
                                     vorhanden */
   PRINT <Brief>
 Else Do
  SAVEAS <temporäre Datei>
  address Pointprogramm
  VERSENDE <temporäre Datei> /* an */ Netzwerkadresse
 end

Until <Datensatz> = <letzter Datensatz>

(* Ende Kasten 1 *)
```

Abb. 1. Ein kurzes Programm zum Probieren



Muschel exquisit

Mit dem neuen Betriebssystem OS 2.0 hat auch die interne Shell Aufpolierung erfahren. Was ist anders?

Einer der unbestreitbaren Vorteile des AmigaDOS ist, daß es dem Anwender grundsätzlich zwei Benutzeroberflächen für das Datei-Handling anbietet: zum einen die mausorientierte "Workbench", zum anderen das textorientierte "Command Line Interface", kurz CLI genannt.

Ab Workbench 1.3 war eine Shell auf der Workbench-Disk vorhanden – aber leider nur zum Nachladen.

Das hat sich mit OS 2.0 gründlich geändert: Sowohl der für Ein- und Ausgabe zuständige Console-Handler als auch die Shell an sich wurden gründlich aufpoliert. Freundlicherweise sind alle Features im Kickstart enthalten und stehen so direkt nach dem Booten zur Verfügung.

Zuerst zum Console-Handler. Angesprochen wird dieser wie gehabt per »CON:«, allerdings sind diverse neue Möglichkeiten integriert. Syntax:

```
CON:x/y/breite/höhe/name/flags
```

Die Positionsangaben funktionieren wie gehabt, mit dem Unterschied, daß zu große Fenster notfalls an die maximale Screen-Größe angepaßt werden.

Ein »CON:0/0/2048/2048/« wird also immer zu einem vernünftigen Ergebnis führen. Interessant wird es bei den Flags.

Zur Verfügung stehen:

CLOSE: Das Fenster bekommt ein Close-Gadget. Wird dies angeklickt, gibt der Handler beim Lesen immer ein EOF (CTRL-/) zurück. Eine eventuelle Shell wird damit – wie bei »ENDCLI« – geschlossen.

WAIT: Das Fenster bleibt auch nach dem Schließen auf, bis eine Taste gedrückt oder ein eventuelles Close-Gadget betätigt wird.

AUTO: Das Fenster bleibt so lange zu, bis zum erstenmal eine Ausgabe vorgenommen wird.

BACKDROP: Das Fenster ist vom Typ "BACKDROP" (es liegt grundsätzlich unter allen anderen Fenstern).

NOBORDER: Das Fenster bekommt keinen Rahmen.

NOSIZE: Das Fenster bekommt kein Sizing-Gadget.

INACTIVE: Das Fenster wird beim Öffnen nicht automatisch aktiviert.

NODRAG: Das Fenster bekommt keinen Dragbar und ist daher nicht verschiebbar.

NODEPTH: Das Fenster bekommt kein Depth-Gadget.
SCREEN: Öffnet das Fenster auf dem angegebenen Public-Screen.
WINDOW: Es wird das angegebene Fenster benutzt (Vorsicht!).
Mit dem Aufruf
NEWSHELL CON:0/0/2048/2048//NODRAG/NODEPTH/NOBORDER/BACKDROP
erhält man prompt eine Shell ohne Rahmen und sonstigen Firlefanz direkt "unten" auf dem Workbench-Screen.

Fenster mit Gedächtnis

Ein weiteres neues Feature ist auf den ersten Blick gar nicht sichtbar: Der Console-Handler speichert die letzten Textausgaben immer mit. Nehmen wir an, Sie geben mit »TYPE« eine Datei im Shell-Fenster (mit maximaler Größe) aus. Nun verkleinern Sie das Fenster – ein Großteil der Ausgabe verschwindet. Ziehen Sie das Fenster wieder auf Maximalgröße – schon ist der Text wieder komplett im Fenster.
Positiver Nebeneffekt: Wird ein "CON:"-Fenster von einem anderen Fenster überlagert, gehen jetzt nicht mehr Unmengen an Chip-Memory für die Speicherung der grafischen Information des verdeckten Bereichs verloren.
Aber es geht weiter: Sie können aus dem Fenster mit der Maus Textbereiche markieren und mit der Tastenkombination [Linke-Amiga-C] "ausschneiden" und in ein anderes Fenster mit [Linke-Amiga-V] wieder einsetzen – wie in einer Textverarbeitung.
Leider wird für diese Schneideoperationen nicht ohne weiteres das »clipboard.device« benutzt. Starten Sie allerdings das Programm »ConClip«, werden ausgeschnittene Texte auch ins Clipboard abgelegt und können von da mit einem geeigneten Programm – zum Beispiel einem Texteditor – wieder eingelesen werden.

Die Shell

Soviel zum Console-Handler. Die Shell an sich hat auch einige nennenswerte Erweiterungen erfahren.
Einige der bisher von Diskette beziehungsweise Festplatte nachgeladenen Shell-Befehle sind jetzt in die Kickstart integriert (Tabelle 1). Dadurch wird die Abarbeitung gerade von Batch-Dateien extrem beschleunigt – und es ist mehr Platz auf der Workbench-Disk frei.
Der bereits in der 1.3-Shell vorhandene Resident-Mechanismus ist ebenfalls vorhanden. Residente Programme werden einmal in den Arbeitsspeicher geladen und bei jedem Aufruf direkt aus dem Speicher ausgeführt. Damit das funktioniert, müssen die Programme "Pure" sein – erkennbar am P-Bit in den Datei-Flags. Zwar ist es möglich, auch nicht-residentfähige Programme resident zu machen. Dann besteht allerdings die Gefahr von unerklärlichen Systemfehlern – Vorsicht damit!
Environment-Variablen werden selbstverständlich unterstützt. Allerdings wird jetzt zwischen globalen und lokalen Variablen unterschieden. Globale Environment-Variablen sind von jedem Programm aus ansprechbar; sie werden als normale Dateien im Verzeichnis »ENV:« gespeichert. Globale Variablen werden mit den Befehlen »SetEnv« und »GetEnv« angesprochen.
Neu sind die lokalen Variablen: Diese sind nur dem entsprechenden Shell-Prozeß bekannt. Angesprochen werden sie mit den Befehlen »Set« und »Get«. Für jeden Shell-Prozeß gibt es drei vordefinierte lokale Variablen:
process: die Nummer des Shell-Prozesses.
RC: der Rückgabewert des letzten aus der Shell aufgerufenen Kommandos.
Result2: die DOS-Error-Number des letzten Befehls. Dieser Wert wird zum Beispiel vom Befehl »Why« benutzt.
Environment-Variablen können Sie jederzeit mit einem vorangestellten "$" in Kommandozeilen benutzen. Zum Beispiel gibt »TYPE $datei« den Inhalt der Datei aus, deren Namen in der Variablen »datei« gespeichert ist.
Der Großteil der Shell-Kommandos befindet sich aber weiterhin auf der Workbench-Disk im Verzeichnis »C:«. Auch dort wurde fleißig aufpoliert, die vielen neuen Features der Befehle einzeln aufzuführen, würde hier zu weit führen. Ganz allgemein haben die Entwickler bei den Kommandos auf einheitliche Parameterfelder und durchgängige Unterstützung von Wildcards (Jokern) geachtet.
Wie auch schon unter 1.2/1.3 läßt sich zu jedem Befehl mit »BEFEHL ?« die Parameterschablone aufrufen. Einige neue Parameteroptionen sind allerdings hinzugekommen:
/N: erwartet eine Ganzzahl.
/M: Angabe beliebig vieler Parameter möglich. Wird oft bei Dateinamen verwendet (siehe zum Beispiel »DELETE«)
/F: Der gesamte Rest der Zeile wird als zusammenhängender Text interpretiert, ob nun Anführungszeichen vorhanden sind oder nicht.
Sicherlich wissen Sie, daß in der Shell die Ausgabe und Eingabe von Programmen mit den Optionen ">" und ">", mit der die Ausgabe an die angegebene Datei angehängt wird. Im Gegensatz zur 1.3-Shell wird die angegebene Datei neu erzeugt, wenn sie nicht vorhanden ist.
Was fehlt noch? Das Piping! Piping bedeutet, daß die Ausgabe eines Befehls als Eingabe eines anderen benutzt wird. Verwirrend? Ein Beispiel: Sie geben mit »LIST« ein Inhaltsverzeichnis aus und wollen mit »TYPE« ein Hex-Listing desselben erzeugen. Eine Möglichkeit wäre

```
LIST >T:TEMPFILE
TYPE T:TEMPFILE OPT H
```

Mit Piping geht es einfacher:

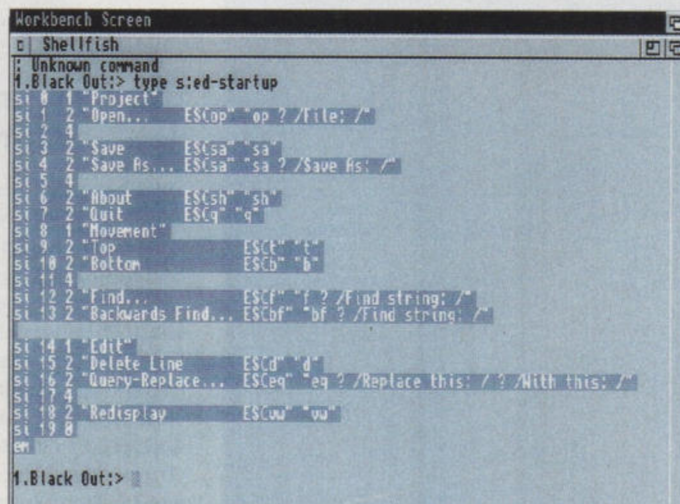
```
TYPE `LIST` OPT H
```

Die Ausgabe des Listings in die einfachen Anführungszeichen (zu finden auf der Taste links neben der "1"; es gibt deren leider noch drei andere) wird also als Datei interpretiert.
Soviel zum kleinen Streifzug durch die Shell 2.0. Wie Sie sehen, hat sich einiges getan – es liegt nun an Ihnen, diese Möglichkeiten zu nutzen!

Interne Shell-Befehle

Alias, Ask, CD, Echo, Else, EndCLI, EndIf, EndShell, EndSkip, Failat, Fault, Get, Getenv, If, Lab, NewCLI, NewShell, Path, Prompt, Quit, Resident, Run, Set, Setenv, Skip, Stack, Unalias, Unset, Unsetenv, Why ❑

Jürgen Stephan (jb)



Texte ausschneiden

Es war einmal ...

Die Geschichte des AmigaDOS

Der Amiga, ursprünglich konzipiert als ultimative Spielkonsole? Inzwischen ist diese Tatsache wohl niemandem mehr ein Geheimnis. Wie kam es dann aber, daß einige Amiga-Modelle inzwischen im High-End-Markt mit einem der fortschrittlichsten Betriebssysteme glänzen?



Schauen wir etwa acht Jahre zurück. Die Joystick-Firma Amiga steuerte aufgrund der Entwicklung des damals noch "Lorraine" geheißenen Flugsimulators der Pleite entgegen und suchte Geldgeber. Der gerade zu Atari gewechselte Jack Tramiel pokerte etwas zu hoch, und das Angebot von Commodore kam genau richtig. Allerdings wollte Commodore keine Spielkonsole – zu diesem Zeitpunkt war der Konsolenmarkt in Europa und den USA gerade am Zusammenbrechen –, sondern einen "richtigen" Computer. Und zwar einen, der alles bisher dagewesene in den Schatten stellen sollte.

Da der Schwenk zum Personal-Computer relativ kurz vor der endgültigen Fertigstellung erfolgte, standen die Entwickler vor eimem Problem: Ein PC braucht vor allem eine Benutzeroberfläche und ein DOS (Disk Operatung System).

Der Kern des Amiga-Betriebssystems stand zu diesem Zeitpunkt schon: der Multitasking-Kern »Exec«, die weitreichenden Möglichkeiten der »graphics.library« und die grafische Benutzeroberfläche »Intuition«.

Moment mal – Spielekonsole und grafische Benutzeroberfläche? Genau das! Die Vermutung, daß die Entwickler der Betriebssystem-Software sich schon länger Gedanken in Richtung "echter" Computer gemacht haben, ist nicht so ganz von der Hand zu weisen.

Allerdings hatte man ein eines nicht bedacht: Eine Spielkonsole braucht weder Diskettenlaufwerk noch Festplatte (CD-ROMs waren damals preislich nicht diskutabel). Also mußte ein DOS her.

Von CAOS zum Chaos

Das hastig ins Leben gerufene Projekt "CAOS" (Commodore Amiga Operating System) konnte bis zur angesetzten Präsentation des Systems nie und nimmer fertiggestellt werden, wie die Entwickler schnell feststellten. Was also tun?

Die Firma Metacomco hatte zu diesem Zeitpunkt ein funktionsfähiges DOS für 68000er-Rechner laufen: »Tripos«. Commodore ergriff den Rettungsanker und kaufte »Tripos« ein. Eine Legende aus dieser Zeit weiß folgendes zu berichten: Der Tripos-Entwickler wurde mit einem Amiga-Prototyp, einem Ausdruck des bisher fertiggestellen Amiga-Betriebssystems und jeder Menge Kaffee in einem Raum eingeschlossen – und kam eine Woche später mit dem fertigen DOS wieder raus. Ein böses Gerücht aus der gleichen Zeit erzählt, dieses würde man dem DOS durchaus anmerken ...

Wie dem auch sei: Das DOS verhält sich im Amiga-Betriebssystem wirklich wie eine Art Fremdkörper. Ist der Großteil des AmigaOS (Operating System) in C oder Assembler geschrieben, wurden »Tripos« und daher auch das AmigaDOS in der Sprache »BCPL« entwickelt. Obwohl C und BCPL denselben Urvater ("B") haben, gibt es bei BCPL ein großes Problem: Der Speicher wird nicht mehr Byte-, sondern Langwortweise angesprochen. Zeichenketten werden per Konvention nicht mit einem Nullbyte beendet, sondern beginnen mit einem Byte, das die Länge angibt.

Die Langwortorientierung bringt auf einem Prozessor wie dem MC68000, der trotz 32-Bit-Architektur weiterhin mit Byte-Adressen arbeitet, natürlich so ihre Probleme mit sich. Sämtliche (Langwort-)Adressen müssen vor Benutzung in Byte-Adressen umgerechnet, das heißt mit vier multipliziert werden. Dieser Effekt ist jedem Programmierer bekannt, der sich schon etwas näher mit dem DOS beschäftigt hat: BCPL-Pointer (deutsch: "Zeiger"), kurz "BPTR", heißen die kleinen Ungeheuer, die schon für unzählige, schwer zu entdeckende Fehler verantwortlich sind.

Kickstart

Nun war, nach einigem Hickhack, AmigaOS 1.0 fertiggestellt. Die Entwickler bekamen allerdings kalte Füße, als ihnen langsam klar wurde, was für ein Sammelsurium an Fehlern und Unzulänglichkeiten sie konstruiert hatten – aus Zeitgründen ging es ja gar nicht anders.

Was nun? Amiga mit halbfertigem Betriebssystem ausliefern oder weiterentwickeln und die Markteinführung verschieben? Es kam die rettende Idee: Warum das Betriebssystem ins ROM tun? Auf Diskette damit, und schon können wir prima "updaten".

So wurden dem Amiga 1000 zu seinen schon gigantischen 256 KByte Hauptspeicher kurzum per Aufsatz weitere 256



KByte RAM spendiert, in die beim Systemstart der Betriebssystemkern geladen wird. Die Diskette, auf der sich das Betriebssystem befand, wurde "Kickstart" getauft – inzwischen hat sich dieser Name als Synonym für das AmigaOS schlechthin etabliert.
Die Markteinführung hatte der Amiga also hinter sich gebracht, nun konnte am Betriebssystem weiterentwickelt werden. Aus 1.0 wurde 1.1, das zwar schon besser, aber trotz allem nicht das Gelbe vom Ei war. Aus der Zeit stammen die schwer ausrottbaren Gerüchte vom absturzanfälligen Amiga.
Der zeitweise ausgelieferte BASIC-Interpreter »ABasiC«, dem Ähnlichkeiten mit dem »ZX-81-Basic« nachgesagt wurden, verschwand von der Extras-Disk und wurde vom nur unwesentlich besseren Amiga-BASIC ersetzt – eine etwas blasse und fußlahme Umsetzung des Microsoft-BASIC für den Apple Macintosh (Meinung des Autors).
AmigaOS 1.1 wurde um einige nützliche Features erweitert und von fast allen wesentlichen Fehlern bereinigt. Das Ergebnis trägt die Versionsnummer 1.2 und diente fast fünf Jahre lang als Basis für alle Amiga-Computer. Im A500 und A2000 wurde das Betriebssystem dann erstmals in einem ROM untergebracht, was A1000-Anwender zu hämischem Grinsen verleitete: Dachten sie doch, mit der ladbaren Kickstart die Zukunft mit allen neuen Betriebssystemversionen gepachtet zu haben (was sich inzwischen als kolossale Fehleinschätzung erwiesen hat: OS 2.0 umfaßt 512 KByte und läßt sich so nicht mehr in das Kickstart-RAM eines A1000 laden).
Nun kam es, daß die Preise für Massenspeicher rapide sanken und so Festplatten auch für einen Rechner wie dem Amiga erschwinglich wurden. Obwohl sich keiner traute, einen Festplatten-Controller speziell für den Amiga zu entwickeln, waren doch preiswerte Festplattensysteme zu bekommen: Pfiffige Bastler nahmen einen XT/AT-Festplatten-Controller (zum Beispiel Omti 5520), blendeten diesen mit wenig Aufwand in den Amiga-Adreßraum ein und emulierten (besseres Wort für "nachmachen") den XT-Bus softwaremäßig – fertig war die Low-Cost-Lösung.
Leider hatten die Amiga-Entwickler eines übersehen: Das AmigaDOS war nur in der Lage, vom Laufwerk »DF0:« zu starten. Ohne Diskette in »DF0:« tat sich gar nichts, ob Festplatte oder nicht. Die bereits oben erwähnten pfiffigen Bastler bastelten weiter und setzten den Festplattentreiber in ein ROM, so daß die Platte direkt nach dem Booten zur Verfügung stand. Trotzdem bootete AmigaDOS stur von »DF0:« weiter. Ein ganz besonders pfiffiger Bastler ging nun her und machte die Festplatte zeitweise zum »DF0:« – endlich war Autoboot möglich.

Reif für die Festplatte

Inzwischen war es 1987; Commodore hatte weiterentwickelt und Pläne für eine völlig neue und überarbeitete Betriebssystemversion, die die ganzen Versäumnisse von



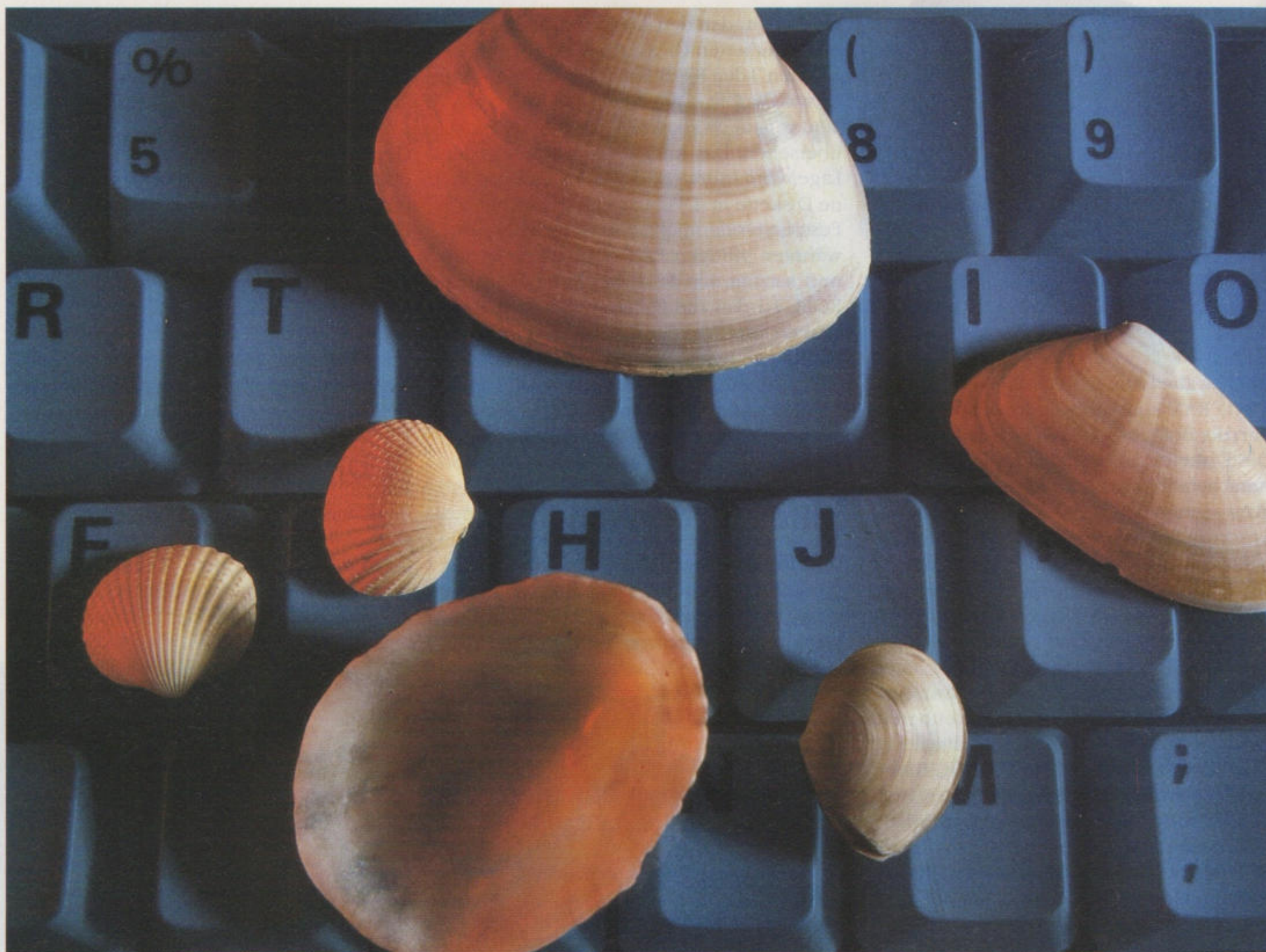
1.0, 1.1 und 1.2 aufholen sollte. Als erstes wurde eine betriebssystemkonforme Möglichkeit zum Booten von beliebigen Geräten (Festplatte, RAM-Disk und so weiter) integriert und die Workbench aufpoliert – heraus kamen die "Shell", verbesserte Bibliotheken und CLI-Befehle.
Nun bekam die Presse schnell Wind von der Sache und zerschrieb das Projekt mit teilweise haarsträubenden Kommentaren über Features und Inkompatibilitäten. Commodore blieb nichts anderes übrig, als die keinesfalls fertige 1.3 zu veröffentlichen – ohne daß auch nur ansatzweise das realisiert wurde, was ursprünglich dafür geplant wurde. Die Kickstart 1.3 ist bis auf die Autoboot-Möglichkeit und die Versionsnummer mit der 1.2 vollkommen identisch.
Die wesentlichen Änderungen wurden auf der Workbench-Disk vorgenommen, aber die Tatsache, daß die Workbench 1.3 auch unter Kickstart 1.2 einwandfrei läuft, unterstreicht die Verwandtschaft der beiden OS-Versionen.
Kickstart 1.3 war also auf dem Markt. Die Gerüchteküche kochte weiter, und es war etwas über eine 1.4 zu hören, bei der alles besser, schöner und neuer sein würde.
Commodore hatte aus den Erfahrungen mit dem 1.3-Debakel gelernt und hielt absolut dicht – die Entwickler machten sich sogar einen Scherz, nahmen eine 1.3-Kickstart, veränderten das Aussehen der Fenster und setzen sie als 1.4-Alpha in Umlauf. Die Amiga-Szene fiel auch prompt auf diese Ente rein.

Die Super-Kickstart

Es dauerte drei Jahre, bis die neue Kickstart, 2.0 geheißen, ans Licht der Öffentlichkeit trat: mit dem Erscheinen des A3000. Allerdings war dem A3000 auch nur eine Beta-Version der Kickstart beigelegt.
Es dauerte ein weiteres Jahr, bis OS 2.0 endgültig auf den Markt kam, aber das Warten hat sich gelohnt: OS 2.0 läßt sowohl für den Anwender als auch für den Programmierer kaum noch Wünsche offen und kann durchaus als ausgereiftestes Betriebssystem für Personal Computer gelten. Und das Schöne daran: Während wesentlich weniger komfortable Betriebssysteme für Intel-Rechner wie »Windows« oder »OS/2« Systemvoraussetzungen wie "mindestens 16-MHz-Rechner mit 4 MByte RAM und SEHR schneller Festplatte" benötigen, läuft OS 2.0 auch auf dem guten alten A1000 mit 7 MHz und 512 KByte RAM noch einwandfrei – und schneller.
Mit OS 2.0 und dem Amiga DOS 2.0 zeigt der Amiga endlich sein wahres "DOS-Gesicht". Wer sich einmal mit der Shell unter OS 2.0 näher beschäftigt hat, wird so schnell nicht mehr von ihr loskommen. Aber nach allem, was man hört, steht auch die Entwicklung bei Commodore nicht still – warten wir es ab ... ❑

Jürgen Stephan (jb)





Einsteigerkurs: Shell und CLI (1)

Ohne Tasten geht es nicht!

Einsteiger, Aufsteiger, Umsteiger aufgepaßt! Wir machen jetzt eine Exkursion durch ein bisher unbekanntes Gebiet des Amiga – dem unerforschten Bereich des "Command Line Interface" und seiner Nachkommen, kurz CLI und Shell genannt. Sie brauchen nicht viel, nur ein bißchen Mut, Ehrgeiz – und saubere Finger für die Tastatur.

Für diejenigen unter Ihnen, die neu hinzugekommen sind, noch eben Erläuterungen zu den Begriffen "Shell" und "CLI". Das "CLI" (Command Line Interpreter) ist ein Zeileneditor. Das bedeutet, daß Eingaben vom Benutzer gleich ausgewertet und ausgeführt werden, solange sie den Vorgaben des Interpreters genügen. Damit überhaupt etwas ausgeführt werden kann, muß ein Programm geladen, ein Parameter eventuell übergeben und eine Bestätigung vom Benutzer gegeben werden. Im Endeffekt sähe das so aus (Achtung! Shell-Window öffnen!):

Eingabe: Hallo Amiga [RETURN]

kurzes Arbeiten des SYS-Laufwerkes, dann

Ausgabe: Unknown Command Hallo

Wir haben dadurch zwei wichtige Informationen gewonnen:

– Das CLI kann Eingaben nicht direkt auswerten, sondern muß erst eine Art Suche beginnen, und

– Eingaben werden zeilenweise abgearbeitet, deshalb ist es (fast!) nicht möglich,

mehrere Kommandos hintereinanderzusetzen.
Beide Angaben lassen sich bei geschickter Ausnutzung bestehender Möglichkeiten allerdings umgehen – wie, das soll Ziel unseres kleinen Kurses werden.
"Nimm einen Amiga ...", hatten die Verkäufer gesagt. "Da brauchst Du keine Tastatur mehr – alles geht mit der Maus...", hatten Sie gesagt. Und nun? Da wird einem erklärt, daß man die Tastatur doch braucht, und sie liegt ja beim Kauf auch bei.
Ganz so schlimm ist es nicht. Im Grunde genommen kann man Anwendungsprogramme auf dem Amiga tatsächlich ohne Tastatur bedienen, die meisten jedenfalls. Trotzdem hat die Tastatur ihre Berechtigung, und zwar nicht nur zum Textschreiben. Wir müssen uns darüber im klaren sein, daß viele Möglichkeiten des Betriebssystems durch "Nur-Maus-Benutzung" brachliegen. Der Grund ist einfach, schauen wir uns doch dazu den Format-Befehl auf unserer Workbench an.

Versteckte Möglichkeiten

»Initialize« heißt der Menüpunkt unter dem Menü »Disk«. Mit diesem Menüpunkt läßt sich eine nagelneue Diskette für Daten vorbereiten oder eine bereits bespielte wieder "erneuern". Wir wollen uns hier nicht mit dem Innenleben von Disketten befassen, nur soviel: Eine Amiga-Diskette hat ja bekanntlich 80 Tracks (von 0 bis 79), die erst angelegt werden müssen. Wählt man den Menüpunkt an, gibt's ein paar System-Requester ("Sind Sie sich wirklich sicher, daß Sie sich sicher sind, daß Sie die Diskette formatieren wollen?"). Das war es aber auch schon, die Diskette wird formatiert – und fertig. Netterweise bekommt Sie noch den Namen "Empty" (wie sinnig!) und einen "Trashcan", einen "Mülleimer" für nicht benötigte Programme draufkopiert.
Also, ich als Amiga-Besitzer muß mir vorschreiben lassen, wie meine Disketten auszusehen haben? Nein, Danke, da versuche ich das lieber über die Shell. Das geht nämlich auch, wenn wir uns in das Verzeichnis »System« bewegen.
Dazu machen wir das Window für die Startdiskette auf und klicken zweimal auf das Shell-Icon. Die Shell müßte sodann erscheinen, und zwar mitten auf dem Bildschirm (nicht besonders nett, siehe auch "Die Shell ist falsch plaziert ..."). Wir geben jetzt mal ein »CD SYS:SYSTEM« und danach »DIR«, um erst das Verzeichnis zu wechseln und dann den Inhalt des Verzeichnisses »System" anzusehen. Dort finden wir durch den Befehl »DIR« (steht für "DIRectory", also "Verzeichnis") das Programm »FORMAT«. Rufen wir es durch Eingabe von »FORMAT« und [RETURN] auf, bekommen wir folgende Zeile zu sehen:
Usage: format DRIVE NAME [NOICONS][QUICK][FFS][NOFFS]
Uff! Da steckt wohl ein wenig mehr dahinter, als es uns die Workbench weismachen wollte. Der Amiga will uns auf diese Art folgendes mitteilen:
»FORMAT« muß mit Parametern angegeben werden. Der erste ist das Laufwerk (DRIVE), wobei das "DRIVE" ebenfalls geschrieben werden muß, also zum Beispiel:
FORMAT DRIVE DF0:
Der Name NAME (auch hier muß der Text "NAME" mit eingegeben werden) dient zum Benennen unserer Diskette:
FORMAT DRIVE DF0: NAME ADALBERT
Diese beiden Parameter sind absolutes Muß, die nachfolgenden jedoch nur Option. »NOICONS« unterläßt das Schreiben des "Trashcan" nach dem Formatieren (spart Bytes). »QUICK« formatiert nicht die komplette Diskette, sondern löscht die vorherigen Einträge und überprüft die Sektoren. »FFS« dient zum Formatieren einer Diskette im "FastFileSystem-Format" (Nutzer von OS 2.0 auf Amiga 500 Plus und Amiga 3000 dürfen sich freuen, denn unter diesem System ist das FFS-Format auch für Disketten nutzbar) und kann den Zugriff auf ihre Disketten vom Rechner aus schneller machen. »NOFFS« schaltet diese Funktion aus. Wie Sie sehen, läßt sich allein beim Formatieren einer Diskette unter der Shell per Tastatur mehr erreichen als mit »Initialize« unter der Workbench. Und wenn das jetzt schon klappt, was muß dann alles noch besser sein?

Ohne Windows geht es nicht

CLI-Kommandos brauchen die Shell, die Shell wiederum braucht ein Window, um sich auf dem Bildschirm zu zeigen. Windows sind ja Ausschnitte des aktuellen Screens, die quasi ein "Eigenleben" haben, also Text darstellen können, ohne daß der aktuelle Screen dadurch "zerstört" wird. Wir als Benutzer der Shell brauchen uns allerdings nicht um das Window zu kümmern, dessen Aussehen wird vom System bestimmt. Änderungen des ursprünglichen Shell-Windows sind allerdings möglich (siehe "Die Shell ist falsch plaziert ..."), wir werden jedoch erst einmal mit eigenen Windows arbeiten. Als ich gerade davon schrieb, daß wir uns nicht um das Shell-Window kümmern müßten, betraf das die Programmierung des Windows, jedoch können wir für Shell-Windows die Größenparameter und den Namen selbst bestimmen.
Um eine Probe aufs Exempel zu machen, starten wir die Shell von unserer Workbench. Das Shell-Window zeigt sich ungefähr in der Mitte des Screens und ist nicht besonders groß. Um nun ein Shell-Window zu bekommen, das den gesamten Screen ausfüllt, geben wir die folgende Zeile ein:

```
NEWSHELL
NEWCON:0/0/640/256/RIESEN-SHELL
```

Übrigens, Sie brauchen innerhalb der Shell keinen Wert auf Groß- und Kleinschreibung zu legen, die Shell akzeptiert alles. Die Befehle hier im Kurs sollen nur schnell für Sie ersichtlich sein.
Mit dem Befehl »NewShell« öffnen wir also ein neues Shell-Window. Geben wir nur »NewShell« ein, wird das neue Window an der gleichen Position wie das vorherige installiert. Durch die Parameter »x1/y1/x2/y2/Shell-Name« werden also die Größe, in unserem Falle der volle Bildschirm, sowie der Name festgelegt.
Was hat es aber mit »NewCon:« auf sich? Der Doppelpunkt verrät es, wir haben es hier mit einem logischen Device zu tun, also einem imaginären Gerät, im Gegensatz zu einem realen Gerät wie zum Beispiel dem zweiten Laufwerk »DF1:«. »NewCon:« ist der Nachfolger von »Con:«, dem Konso-

Kleiner Kursfahrplan

Dieser Kurs richtet sich vor allem an Einsteiger, die den Amiga gerade in Besitz genommen haben, sowie an diejenigen, die bisher noch keine Berührung mit Shell und CLI hatten. In unserem Kurs werden wir den praktischen Umgang mit Shell und CLI lernen sowie Erleichterungen in der Arbeit austüfteln. Außerden werden wir bestimmte Befehle näher untersuchen, die einiges an Funktionen zu bieten haben. Des weiteren lernen wir den Umgang mit Batch-Dateien.
Teil 1: Was ist die Shell? – Con- und NewCon-Handler – Öffnen von Shell-Windows – Alias-Funktionen – Besserer Format-Aufruf
Teil 2: Das AmigaDOS – Besondere Shell-Befehle – Erstellen einer Start-Diskette – Die »Startup-Sequence« – Was sind "Tasks"? – Steuerzeichen für die Shell
Teil 3: Batch-Dateien erleichtern die Arbeit – Komfortable Kommandofolgen – FD-Software für Shell-Liebhaber – Batch-Aufrufe von der Workbench

len-Device. Eine Konsole ist ein Eingabegerät, damit ist unsere Tastatur gemeint. Damit nun die Eingaben von der Tastatur überprüft und ausgewertet werden können, benötigt man ein "Programm", welches diese Arbeit übernimmt. Dabei kann es sich allerdings nicht um ein normales Programm handeln, denn die Eingaben müssen vom System überwacht werden, die Tastatur muß ja dauerhaft im Einsatz sein.

Wenn wir unsere SYS-Diskette untersuchen, dann finden wir das Verzeichnis »L«, in dem wir den "NewCon-Handler" finden. Warum aber nicht den "Con-Handler"? Der ist im System bereits integriert, genauer gesagt, er befindet sich schon in den Bausteinen des Amiga, die das Betriebssystem enthalten. Und hier finden wir gleich den ersten Unterschied zwischen den verschiedenen Betriebssystemen:

– Kickstart 1.2 enthält nur den "Con-Handler" im ROM.

– Kickstart 1.3 bekam zusätzlich den "NewCon-Handler" ins L-Verzeichnis, der vom System nachgeladen werden muß. Erst dadurch wird das CLI zur Shell.

– Kickstart 2.0 bekam ein völlig neues CLI, hier enthält der "Con-Handler" gleich die Verbesserungen, die unter OS 2.0 entstanden.

Kurzum, wollen wir die Shell auf andere Disketten bringen, müssen wir den "NewCon-Handler" auch berücksichtigen – neben ein paar anderen wichtigen Einzelheiten, auf die wir aber erst später kommen.

Was wir bis jetzt wissen, ist folgendes:

– Die Shell ist ein verbessertes CLI, das "Zubehör" benötigt, um gestartet werden zu können. Sie enthält wesentlich mehr Möglichkeiten, als die Workbench uns bieten kann.

– Um die Shell aufzurufen, genügt ein einfaches Klicken des Shell-Icons. Über die Shell selbst lassen sich weitere Shells öffnen, deren Größe und Namen wir selbst bestimmen können.

Die Alias-Funktion

Was wir im heutigen Teil noch lernen wollen, ist, einen Funktionsaufruf zu schreiben und ihn als Alias-Kommando unterzubringen. Dazu nehmen wir den Format-Befehl aus dem Verzeichnis »System«.

Ich habe vorhin schon einmal kurz die Alias-Funktion erwähnt. Sie läßt eine Abarbeitung von kompletten Kommandofolgen in Kurzschreibweise zu. Gesteuert wird dieses durch die »Shell-Startup«, einer Datei im System-Verzeichnis »S«, in der die Kurzform sowie die abzuarbeitende Kommandofolge eingetragen werden müssen. Die Shell, oder besser der "NewCon-Handler", übernimmt es, daß die Kurzformen als Befehlssequenz erkannt werden, in dem er die »Shell-Startup« nach der Kommandofolge "absucht".

Schauen wir uns noch einmal den Format-Befehl aus dem Verzeichnis »System« an. Wie wir vorhin feststellen konnten, ist der Funktionsaufruf von der Shell wesentlich umfangreicher als das eintönige »Initialize« von der Workbench. Zudem bekommt man gratis immer noch den Mülleimer, den "Trashcan" mit auf die Diskette gepackt, den man so nötig hat wie einen zweiten Blinddarm. Er nimmt Speicherplatz von der Diskette, den wir noch gut gebrauchen könnten, und außerdem: Wenn ich ein Programm löschen will, dann will ich auch. Also: Weg mit dem Mülleimer. Das geht aber nur, wenn man so formatiert. Unsere Anforderung an das Kommando soll so aussehen:

1) Formatieren einer Diskette im Standard-Format.

2) Das Formatieren soll ohne Rückfrage "Ok to initialize Volume XXX (all Data will be erased)?" stattfinden. Wir müssen uns natürlich im klaren sein, daß bei einem vorschnellen Aufruf die falsche Diskette formatiert wird.

3) Das Anzeigen der formatierten Tracks soll in einem eigenen Window stattfinden.

4) Um Fehlaufrufe zu vermeiden, erzeugen wir ein Alias-Kommando.

Die korrekte Zeile für den Format-Befehl ohne Icons wäre damit

```
FORMAT DRIVE DFX: NAME XXXXX
NOICONS
```

Da unser neuer Format-Aufruf vorerst nur einen Namen verwalten kann, wählen wir einen, der auf eine neuformatierte Diskette hinweist, zum Beispiel "Neue_Disk". Sie können natürlich auch "Empty" nehmen – das zeugt aber von wenig Phantasie ...

Mit unseren Vorgaben würde die Zeile jetzt so aussehen:

```
FORMAT DRIVE DF1: NAME Neue_Disk
NOICONS
```

Besitzer eines einsamen DF0:-Laufwerks müssen natürlich auch "DF0:" in die Zeile einsetzen, aber immer daran denken: Vor dem Aufruf nachschauen, was tatsächlich im Laufwerk sitzt.

Mit dieser Zeile läßt sich schon was anfangen, aber wir wollen ja mehr. Deshalb schauen wir uns erst mal die »Shell-Startup« aus dem Verzeichnis »S« genauer an. Dazu benutzen wir den Editor »ED« aus dem Verzeichnis »C« und rufen die »Shell-Startup« auf mit

```
ED S:SHELL-STARTUP
```

Da das Verzeichnis »S« auch als logisches Laufwerk »S:« existiert, können wir die Eingabe wie eben gezeigt verkürzen. Kurz darauf sehen wir die Datei "in natura", meine eigene sieht zur Zeit so aus:

```
Prompt "%N.%S> "
alias endshell endcli
alias clear echo "*E[0;0H*E[J"
alias reverse echo
"*E[0;0H*E[41;30m*E[J"
alias normal echo
"*E[0;0H*E[40;31m*E[J"
alias mc execute s:asm1.BAT
alias ta execute s:point.BAT
alias DP excute s:DPaint.BAT
alias phone execute s:ncomm.BAT
```

Ich gebe zu, das ist nicht gerade berauschend, aber hilfreich. Gehen wir die Zeilen mal durch: Die Zeilen 1 bis 5 sind vom System vorgegeben und haben folgende Effekte:

Zeile 1: Prompt der Shell setzen. Das "Prompt" ist das Eingabezeichen, in diesem Fall die Task-Nummer (mehr dazu im dritten Teil), ein ".", danach der aktuelle Pfad und ein ">" plus Leerzeichen. "%N" ist das Steuerzeichen für die Tasknummer, "%S" für den aktuellen Pfad. Über diese Steuerzeichen werden wir uns auch noch unterhalten. Komplett sieht das zum Beispiel so aus "1.WB_1.3:> ".

Zeile 2: Hier wird der Befehl »Copy« mit einem Übergabeparameter sowie einem Befehlsparameter aufgerufen. Übergabeparameter werden beim Aufruf der Kurzform mit eingegeben. Wie das geht, erfahren wir später.

Zeile 3: Weil die Shell keinen eigenen Befehl zum Beenden kennt, wird einfach der CLI-Befehl »EndCLI« "umbenannt". So kann der Shell-User jetzt "EndShell" eingeben, ausgeführt wird immer noch »EndCLI«.

Die Zeilen 4 und 5: Hier werden Steuer-Sequenz-Eingaben vereinfacht. Ich ärgere Sie jetzt mal: Geben Sie in der Shell doch einmal ein

```
echo "*e[8m"
```

und versuchen Sie, aus dieser Falle wieder herauszukommen, ohne einen "Reset" zu machen. Ich gebe Ihnen einen Tip: Die Zahl "0" macht's ...

In den Zeilen 10 bis 13 habe ich Batch-Dateien (Kommandofolgen), die bei mir im S-Verzeichnis stehen, durch Kurzform aufrufbar gemacht. Zum Beispiel Zeile 11: Dort steht das DFÜ-Programm »THE ANSWER«, repräsentiert durch die Kommandofolge »Point.BAT« ("BAT" ist für mich die Abkürzung für "Batch" und zeigt mir, daß hier eine Kommandofolge und kein Pro-

gramm steht. Diese Kommandofolge im S-Verzeichnis sieht so aus

```
stack 50000 ; Bei TA: Je höher, desto
; besser
cd point: ; "Umschalten" auf die
; HD-Partition "POINT:"
run TheAnswer ; Abkoppeln des Pro-
gram-
; ms vom CLI
cd sys: ; Nach Ende wieder zurück in
; die Boot-Partition
```

Durch das DOS-Kommando "Execute" kann ich eine Kommandofolge abarbeiten, mit der Shell-Startup-Kurzform "ta" rufe ich also den Befehl »Execute« auf und übergebe ihm direkt als Parameter den Namen der Datei, die ich starten will.

Kommen wir zurück zu unserem Format-Befehl. Wir wollen ihn automatisch starten, ohne die Rückfrage vom System, also müssen wir das DOS-Kommando »RUN« benutzen, um den Befehl von unserem Shell-Window "loszukoppeln". Wir gehen jetzt mit den Cursor-Tasten an das untere Ende der Datei (endet die "Cursor-Fahrt" am Ende der letzen Zeile, drücken wir einfach [Return]) und geben ein

```
FORMAT >CON:0/0/500/50/FMT_DF1
DRIVE DF1
: NAME Leer_Disk NOICONS
```

Drücken Sie aber bitte noch nicht [Return], denn die Zeile ist noch nicht fertig. Ich will erst etwas zu dem Parameter nach "FORMAT" sagen. Die Angaben "0/0/500/50/FMT_DF1" dürften Ihnen schon bekannt sein, es handelt sich um die Größenangaben für ein Window.

Dieses Window muß erst erzeugt werden, wir sehen aber keinen "NewShell"- oder "NewCon"-Befehl. Brauchen wir auch nicht, denn wir sagen durch das Zeichen ">" einfach, daß sich der Befehl »Format« NUR in einem durch den Con-Handler erzeugten Window mit den nachfolgenden Größen "austoben" darf. Wir haben somit eine Befehlsumleitung gemacht. Wie das genauer geht, erkläre ich Ihnen beim nächstenmal, nur soviel noch: Wenn wir dem Befehl jetzt ein »Run« voranstellen, so können wir weitermachen in der Shell, obwohl eine Diskette formatiert wird – hoch lebe das Multitasking!

Die vollständige Zeile muß daher so aussehen:

```
ALIAS FMT_1 RUN FORMAT
>CON:0/0/500/50/
FMT_DF1 DRIVE DF1: NAME Leer_Disk
NOICO
NS
```

Durch den Befehl »ALIAS« wird festgelegt, daß die Eingabe von "FMT_1" und einem abschließenden [Return] die Formatierung einer Diskette in Laufwerk 2 in einem eigenen Window vornimmt, ohne Sicherheitsabfrage und ohne Kopieren des Trashcan. Wir speichern die Datei ab, indem wir die Tasten [Esc] und danach [X] drücken.

Jetzt müssen wir einen "Reset" auslösen und den Rechner neu über die Workbench starten. Wenn wir jetzt die Shell öffnen, eine Leerdiskette in »DF1:« einlegen, die Kurzform "FMT_1" plus [Return] eingeben, dann sollte sich das Window öffnen, die Diskette formatiert werden, wobei die Angaben über die Tracks im Window stattfinden, während wir durch Maus-Klick in unsere Shell zurückkehren und uns den Inhalt der Disk in Laufwerk 1, also »DF0:« ansehen. Ist die Disk in »DF1:« fertig formatiert, muß auch das Window wieder schließen.

Die Aufgabe

War doch einfach, oder? Mit Hilfe der Alias-Kommandos läßt sich eine Menge Arbeit in der Shell erleichtern.

Was aber machen diejenigen, die kein zweites Laufwerk haben? Das soll Aufgabe für alle sein, die diesen Kurs mitverfolgen.

Die Lösung ist simpel, wenn sie sich alle Schritte vor Augen führen und das beherzigen, was ich zu Anfang geschrieben habe. Hier trotzdem ein paar Anhaltspunkte:

– Bei nur einem Laufwerk darf das Formatieren nicht automatisch beginnen!

– Der Eintrag für das Laufwerk muß »DF0:« sein und nicht »DF1:«.

Wenn Sie alles begriffen haben, was wir hier im Kurs besprochen haben, dann dürfte Ihnen die Antwort nicht schwerfallen. Aber Sie haben immerhin vier Wochen Zeit – bis zur nächsten AMIGA DOS. Und dann erkläre ich Ihnen, woher unsere Zeitschrift ihren Namen hat – auch wenn er falsch geschrieben zu sein scheint.

Bis dahin wünsche ich Ihnen viel Vergnügen in der Shell – und beherzigen Sie den Ratschlag, den ich Ihnen im Info gab. ❑

(jb)

AMIGA-DOS-Info

Wichtig! Verwenden Sie für Ihre Experimente in der Shell NIE die Original-Disketten der Workbench und der Extras-Disk, sondern machen Sie sich genug Kopien. Gerade bei unserem Beispiel mit der Format-Routine ist dies wichtig.

Die Shell ist falsch plaziert ...

... behaupte ich einfach mal. Es stört mich jedesmal, wenn ich ein Shell-Fenster öffne, dieses auf einen Platz mitten im Bildschirm plaziert wird und somit wichtige Windows einfach in den Hintergrund drängt. Das ewige Anklicken des "WindowToFront-Gadgets" geht mir auch ein wenig auf die Nerven.

Dieser Effekt tritt eigentlich nur dann zu Tage, wenn wir die Shell das erste Mal öffnen, jede weitere Shell kann ja durch Parameter an jeden beliebigen Platz des Screens gesetzt werden. Um trotzdem Einfluß nehmen zu können, brauchen wir ein Programm, das ein wenig in anderen Programm "herumstochern" kann, der Fachmann spricht von "patchen". Ich greife in solchen Fällen auf das Shareware-Programm »File-Master« von Roger Fischlin zurück. Es handelt sich dabei um einen File-Editor, der Programme ins RAM lädt und wie ein Texteditor Eingaben (und somit Änderungen) innerhalb der Files zuläßt. Nach dem Start des »File-Master« und den Infos des Autoren laden wir über einen Requester aus dem Verzeichnis »SYS:SYSTEM/« das Programm »CLI«. »SYS:« steht dabei immer für das Boot-Device, also Workbench-Diskette oder Workbench-Partition der Festplatte.

Das Programm »CLI« wird ins RAM geladen, danach stellen wir in der Menüleiste des »File-Master« erst einmal nur die ASCII-Zeichen von 32 bis 125 ein, um dem Zeichenwirrwarr Herr zu werden. Durch viermaliges Drücken des Feldes "Forward" (Vorwärtsblättern der Sektoren) erreichen wir den Teil des Programms, der für uns interessant wird. Hier finden wir Eintragungen für die Window-Größe des CON- und NEWCON-Fensters. Beim CON-Fenster können wir jetzt mit einem Maus-Klick den Edit-Cursor setzen und eigene Werte eintragen, zum Beispiel "0/170/560/85/My Shell" mit ein paar "Füllern", um die restlichen Buchstaben des ersten Shell-Namens zu löschen. (Vorsicht! Keine benötigten Daten überschreiben!) Das neue CLI-Window macht jetzt an den Koordinaten 0,170 (linke obere Ecke: x,y) und 560,255 (rechte untere Ecke: x,y) mit dem Namen "My Shell" auf. "255" deshalb, weil die Window-Daten für den zweiten y-Wert zum ersten addiert werden müssen (170+85=255, der PAL-Screen der Workbench kann 640 Bildpunkte in der Horizontalen und 256 in der Vertikalen benutzen, man spricht von "Med-Res", mittlerer Auflösung).

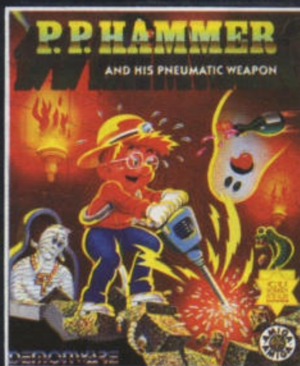
Nach Schreiben der neuen Werte in den Sektor mit "Write" speichern wir das Programm »CLI« wieder als Systemdatei ab. Beim nächstenmal sitzt das CLI links unten in der Ecke und stört nicht die freie Sicht. Das Ganze ist zwar mehr kosmetischer Natur, zeigt aber, daß das CLI ebenfalls "nur" ein Programm ist, das man auf diese Art ändern kann.

AMIGA DOS SHOP



Erkunden Sie 270 Sternensysteme mit 800 Planeten und sieben außerirdischen Rassen, einschließlich Ausbildung von Offizieren und Trickfilmgrafik mit Planetenrotation.

49,95 DM*



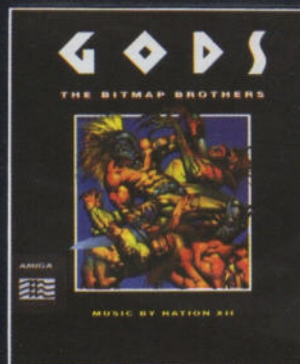
Sein einziges Werkzeug ist der Preßlufthammer. Mit ihm kann er versteckte Steine zertrümmern, um Schätze und geheime Levelausgänge zu entdecken. Ein wochenlanges Vergnügen mit diesem Jump'n'Run-Game.

54,— DM*



Das bekannte Fantasy-Abenteuer jetzt 50 % größer mit sechs Städten und 25 scrollenden Labyrinthen. Genau das Richtige für ausgefuchste Abenteuer.

39,95 DM*



Sie sind Herkules, der in diesem neuartigen Action-Adventure die legendäre Stadt durchstreift, um den Schatz zu finden, der Unsterblichkeit verspricht. Ein unbesiegbarer Krieger auf der Suche nach Abenteuern – eine Herausforderung der Götter.

69,— DM*

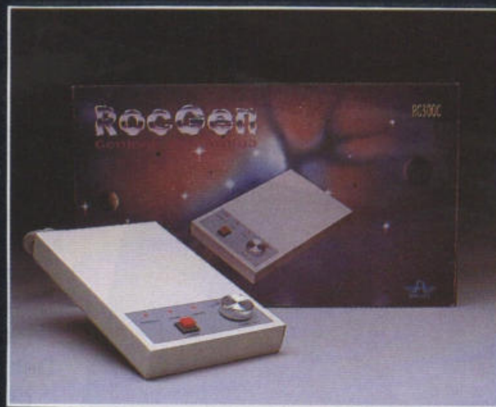
ACHTUNG! Super-Hardware!! Jetzt zugreifen!!

Die Amiga-Maus: RocTec

Maus mit zwei Tasten, hohe Auflösung, sehr gutes Handling, mit Mikroschaltern

nur 35,— DM*





Interface: mischt Videobilder mit Computergrafik oder -text. Überlagerungseffekte, weiches und stabiles Fading, automatischer Video-Durchlauf, voll zugängliches, benutzerfreundliches Bedienungsfeld, Stromversorgungsauswahl extern/intern.

nur 229,— DM*

* Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 4,— DM bzw. für das Ausland 6,— DM Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · W-3440 Eschwege

DMV Daten- und Medienverlag

Fortsetzung von S. 65

Während die Audio-Ausgabe spielt, kann man weiterhin mit Mausklick leisestellen, zum Beispiel wenn das Telefon klingelt. Dies ist das einzige Feature, das die standardmäßige Audio-Steueroberfläche nicht bietet. Man könnte es natürlich noch aufwendiger programmieren, daß man eine beliebige Lautstärke beispielsweise über Schieberegler einstellen kann, aber ich war schon mit diesem Umschalten zufrieden.

Wenn ein Stück im "Einzeln-Modus" zu Ende ist oder man "Abbruch" angeklickt hat, wird das Abspielen mit »AbortIO()« und dann dem Befehl »CD.STOPPLAY« beendet und zur Marke "Eingabe:" zurückgesprungen.

Während dieses Abspielmodus ist das Programm ja normalerweise unbeschäftigt. Damit im Multitasking aber nicht ständig Rechenzeit durch das vergebliche Abfragen der Tasten und der Maus vergeudet wird, ist mit dem Befehl »Delay()« aus der »dos.library« dafür gesorgt, daß diese Abfragen nur wenige Male je Sekunde stattfinden und der Rechner ansonsten uneingeschränkt anderen Programmen zur Verfügung steht, zum Beispiel der genannten Grafikdemo. Ansonsten würde man sich des Vergehens des "Busy Waiting" schuldig machen. Auch im Programmteil "Eingabe" gibt es so eine Delay-Verzögerung, allerdings kürzer, da hier ja öfter mit Eingaben zu rechnen ist und diese möglichst prompt verarbeitet werden sollen. (In der Programmiersprache C löst man dies noch eleganter mit Wait()-Befehlen, die das Programm exakt bei einer Eingabe wieder "aufwecken", das geht aber unter Basic nicht.)

Hier noch Tips, wie man eine entsprechende Diskette anfertigt:

1. Eine Kopie der Original-1.3-Workbench anfertigen.
2. Die Diskette in "CD" umbenennen.
3. Mit dieser Diskette neu starten.
4. Von dieser Diskette von der Shell aus die Fonts löschen, damit man Platz gewinnt: »delete CD:fonts all«.
5. In »Utilities« alles bis auf »Clock« und »Calculator« löschen.
6. Von der 1.3-Extras-Diskette AmigaBASIC in das Hauptverzeichnis von »CD:« kopieren.
7. Von derselben Diskette aus der BASIC-Demos-Schublade die Dateien »dos.bmap« und »exec.bmap« in das Verzeichnis »libs« auf »CD:« kopieren.
8. Das Programm »CD-Steuer1« inklusive seinem Piktogramm auch dorthin kopieren.
9. CDTV-Player damit neu starten.
10. Audio-CD einlegen.
11. Per Maus Diskette öffnen und durch Doppelklick "CD-Steuer1" starten.
12. Nachdem eine Musik gestartet ist, irgendeine schöne Grafikdemo starten. Das Fenster von »CD-Steuer1« kann mitsamt dem Workbench-Schirm ganz nach unten an den Rand gezogen werden.

Dieses noch kleine und relativ primitive Programm macht hoffentlich noch mehr Leuten Mut, sich kreativ mit diesem zukunftsträchtigen Medium des CDTV-Systems zu beschäftigen. ❑

Dr. Peter Kittel (jb)

```
REM CD-Steuerl, P. Kittel, Ffm 19.11.91, 24.11.91
REM aufbauend auf:
REM  »ScreenPrint«
REM  Carolyn Scheppner   CBM USA 04/86
REM
REM  ***  Haupt-Programm  ***
REM
'
WINDOW 1," Peters CD-Audio-Steuerung 1 ",(50,0)-(390,2
7),22
DEF FNnn$(x)=RIGHT$("  "+STR$(x),2)
AlreadyDeclared = 0
'
'                     io_Offset in Ofs&, io_Length in Lae&
, io_Data in Dat&
CD.RESET     =1
CD.MOTOR     =9
CD.ISROM     =37 ' alles 0, io_Actual=0(kein ROM)
CD.PLAYTRACK=43 ' io_Offset=StartTrack, io_Length=Stop
Track, io_Data=0
CD.PAUSE     =52 ' io_Offset=0, io_Length=1(Pause) oder
 0(weiter), io_Data=0
CD.STOPPLAY =53
CD.TOCMSF    =49 ' io_Offset=Track, io_Length=TrackCoun
t, io_Data=TOCArray
                 ' 0. Eintrag in TOCArray: 1. Tracknr.,
 letzte Tracknr.
CD.FADE      =57 ' io_Offset=Laut (max $7FFF),io_Length
=steps(1/75 s),Dat=0
CD.SUBQMSF  =51
'
' Ist es eine Audio-CD?
CDTest:
PRINT :PRINT " Ist es eine Audio-CD?"
command%=CD.ISROM:Ofs&=0:Lae&=0:Dat&=0:kan=0
GOSUB CD
IF Act& THEN
  PRINT " Nein!"
  PRINT " Bitte einlegen und Taste";
  WHILE INKEY$="":WEND
  GOTO CDTest
  END IF
'
' Inhalt
PRINT " CD-Audio-Disc mit";
command%=CD.TOCMSF:Ofs&=0:Lae&=nTOC:Dat&=TOCArray&:kan
=0
GOSUB CD
PRINT PEEK(TOCArray&+3);" Spuren";
max=PEEK(TOCArray&+3)
'
' Normale Lautstärke:
laut&=&H7FFF
command%=CD.FADE:Ofs&=laut&:Lae&=150:Dat&=0:kan=1
GOSUB CD
Delay 100
'
DIM ms$(max)
mi9=PEEK(TOCArray&+5):se9=PEEK(TOCArray&+6)
mil=0:sel=0
FOR i=1 TO max
  mi0=mil:se0=sel
  IF i<max THEN
    mil=PEEK(TOCArray&+i*8+5+8):sel=PEEK(TOCArray&+i*8
+6+8)
            ELSE
    mil=mi9:sel=se9
    END IF
  mi0=mil-mi0:se0=sel-se0
  IF se0<0 THEN se0=se0+60:mi0=mi0-1
  ms$(i)=        RIGHT$("0"+MID$(STR$(mi0),2),2)+":"
  ms$(i)=ms$(i)+RIGHT$("0"+MID$(STR$(se0),2),2)
  NEXT
'
n=1:i=0:cr$=CHR$(13):e$=CHR$(27):leise=0:einz=0
'
Eingabe:
  WINDOW 1
  CLS:d=1
  PRINT "Spur: +    -  +10    Länge:          Ende"
  PRINT "Max.:           -10     Zeit:          Leise"
  PRINT "Spur  0  bedeutet Ende                Einzeln";
  LOCATE 1,9:PRINT FNnn$(n);:LOCATE 2,9:PRINT FNnn$(ma
x);
  GOSUB eneu
  IF i<>0 AND einz=0 THEN
    n=n+1:IF n>max THEN n=1
    GOSUB nneu
                    ELSE
    IF n>max THEN n=max
    GOSUB nneu
    i=1
    WHILE i
      a$=INKEY$
      IF a$=e$  THEN i=0:n=0
      IF a$=cr$ THEN i=0
      IF a$>="0" AND a$<="9" THEN
        n1$=n2$:n2$=a$:n=VAL(n1$+n2$):d=0
        IF n>max THEN n1$="":n=VAL(n2$)
        GOSUB nneu
        END IF
      m=MOUSE(0)
      IF m THEN
        x=MOUSE(1):y=MOUSE(2):d=0
        WHILE MOUSE(0):WEND
        IF y<9 THEN
          IF x< 33 THEN i=0
          IF x> 47 AND x< 56 THEN n=n+ 1:GOSUB nneu
          IF x> 87 AND x< 96 THEN n=n- 1:GOSUB nneu
          IF x>111 AND x<136 THEN n=n+10:GOSUB nneu
          IF x>287 AND x<320 THEN n=0:i=0
          END IF
        IF y>8 AND y<17 THEN
          IF x>111 AND x<136 THEN n=n-10:GOSUB nneu
          IF x>287 AND x<328 THEN leise=1-leise:GOSUB
lneu
          END IF
        IF y>16 AND y<25 THEN
          IF x>279 AND x<336 THEN einz =1-einz :GOSUB
eneu
          END IF
        END IF
      IF d THEN Delay 10
      WEND
    IF n>max THEN n=max
    IF n<1 THEN cleanupl
    END IF
```

```
      '
      command%=CD.PLAYTRACK:Ofs&=n:Lae&=0:Dat&=0:kan=0
      IF einz=0 THEN Lae&=max+1
      GOSUB CD
      '
      LOCATE 1,36:PRINT "Abbruch";
      '
      i=3:wart&=150&
      WHILE i>0
        Delay wart&:wart&=20&
        a$=INKEY$:IF a$=e$ THEN i=0
        m=MOUSE(0)
        IF m THEN
          x=MOUSE(1):y=MOUSE(2)
          WHILE MOUSE(0):WEND
          IF y<9              AND x>279 AND x<336 THEN i=0
          IF y>8  AND y<17 AND x>287 AND x<328 THEN leise=1-leise:GOSUB lneu
          IF y>16 AND y<25 AND x>279 AND x<336 THEN einz = 1-einz :GOSUB eneu
        END IF
        '
        command%=CD.SUBQMSF:Ofs&=0:Lae&=0:Dat&=cdStatus&:kan=1
        GOSUB CD
        '
        LOCATE 2,29
        PRINT RIGHT$("00"+MID$(STR$(PEEK(cdStatus&+ 9)),2),2);":";
        PRINT RIGHT$("00"+MID$(STR$(PEEK(cdStatus&+10)),2),2);
        liesn=PEEK(cdStatus&+2)
        IF PEEK(cdStatus&)<>17 OR (einz AND (liesn<>n)) THEN
          i=i-1:IF i=0 THEN i=-1
                                                    ELSE
          IF i>0 THEN i=3
        END IF
        IF einz=0 AND liesn=(n+1) THEN n=n+1:GOSUB nneu
      WEND
    '
    AbortIO(ioRequest&(0))
    WaitIO( ioRequest&(0))
    command%=CD.STOPPLAY:Ofs&=0:Lae&=0:Dat&=0:kan=0
    GOSUB CD
    '
    GOTO Eingabe
  '
  '
nneu:
IF n<0   THEN n=0
IF n>max THEN n=max
a$=FNnn$(n)
n1$=LEFT$(a$,1):IF n1$=" " THEN n1$="0"
n2$=RIGHT$(a$,1)
LOCATE 1,9 :COLOR 3:PRINT n1$;n2$;:COLOR 1
LOCATE 1,29:PRINT ms$(n);
RETURN
'
'
lneu:
IF leise THEN
  COLOR 3:laut&=3000
        ELSE
  COLOR 1:laut&=&H7FFF
  END IF
LOCATE 2,37:PRINT "Leise";
COLOR 1
command%=CD.FADE:Ofs&=laut&:Lae&=150:Dat&=0:kan=1
GOSUB CD
Delay 100
RETURN
'
'
eneu:
LOCATE 3,36
IF einz THEN PRINT "Einzeln"; ELSE PRINT "Endlos ";
RETURN
'
'
'
'
CD:
'
REM  Beim ersten Aufruf dieser Routine
REM   sind die Funktionen aus der Exec-Library
REM   mit Ergebniswerten zu deklarieren:
'
IF AlreadyDeclared = 0 THEN
  DECLARE FUNCTION AllocSignal%()  LIBRARY
  DECLARE FUNCTION AllocMem&()     LIBRARY
  DECLARE FUNCTION FindTask&()     LIBRARY
  DECLARE FUNCTION DoIO&()         LIBRARY
  DECLARE FUNCTION CheckIO()       LIBRARY
  DECLARE FUNCTION OpenDevice&     LIBRARY
  AlreadyDeclared = 1
  nTOC=100
  '
  LIBRARY "exec.library"
  LIBRARY "dos.library"
  '
  REM *** CreatePort ***
  '
  ClearPublic& = 65537&
  RAMsize&=8*nTOC+2*(40+64)+40
  msgPort& = AllocMem&(RAMsize&,ClearPublic&)
  IF msgPort& = 0 THEN
    PRINT "Speicher nicht allokierbar."
    GOTO cleanup4
    END IF
  '
  msgPort&(0)   =msgPort&
  msgPort&(1)   =msgPort&(0)+40&
  ioRequest&(0)=msgPort&(1)+40&
  ioRequest&(1)=ioRequest&(0)+64&
  cdStatus&     =ioRequest&(1)+64&
  TOCArray&     =cdStatus&+40&
  '
  FOR ip=0 TO 1
    sigBit%(ip) =  AllocSignal%(-1)
    POKE (msgPort&(ip) +  8), 4 'Type=NT_MSGPORT
    POKE (msgPort&(ip) +  9), 0 'Priority 0
    portName$(ip) = "MyCDPort"+CHR$(ip+49)+CHR$(0)
    POKEL(msgPort&(ip) + 10), SADD(portName$(ip))
    POKE (msgPort&(ip) + 14), 0 'Flags
    POKE (msgPort&(ip) + 15), sigBit%(ip)
    sigTask& = FindTask&(0)
    POKEL(msgPort&(ip) + 16), sigTask&
    '
    CALL AddPort(msgPort&(ip))   'Port hinzufuegen
    '
    REM  *** CreatExtIO ***
    '
    POKE (ioRequest&(ip) +  8),5 'Type=NT_MESSAGE
    POKE (ioRequest&(ip) +  9),0 'Priority 0
    POKEL(ioRequest&(ip) + 14),msgPort&(ip)
    '
    REM  *** Open  CDTV.Device ***
    '
    devName$ = "cdtv.device"+CHR$(0)
    pError& = OpenDevice&(SADD(devName$),0,ioRequest&(ip),0)
    IF pError& <> 0  THEN
      PRINT "CDTV nicht ansprechbar."
      GOTO cleanup2
      END IF
    NEXT
  END IF
' Ende Initialisierung
'
' Eigentlicher Aufruf:
'
IF kan=0 OR kan=1 THEN
  POKEW(ioRequest&(kan) + 28), command%
  POKE (ioRequest&(kan) + 30), 0 ' Flags
  POKEL(ioRequest&(kan) + 36), Lae&
  POKEL(ioRequest&(kan) + 40), Dat&
  POKEL(ioRequest&(kan) + 44), Ofs&
  '
  IF command%=CD.PLAYTRACK THEN
    SendIO(ioRequest&(kan))
    ioError&=0
                              ELSE
    ioError& = DoIO&(ioRequest&(kan))
    END IF
                          ELSE
  ioError&=99999&
  END IF
'
IF ioError& <> 0 THEN
   PRINT "CDTV.Device-Fehler =";ioError&
   IF ioError&=-6        THEN
     PRINT "Keine Disc eingelegt"
     PRINT "Bitte einlegen und Taste";
     WHILE INKEY$="":WEND
     END IF
   IF ioError&=-8        THEN PRINT "Kein Inhaltsverzeichnis verfügbar."
   IF ioError&=999996& THEN
     PRINT "Falsche Kanalnr.";kan;"statt 0 oder 1."
     PRINT "Bitte Taste für Ende.";
     WHILE INKEY$="":WEND
     GOTO cleanup1
     END IF
   PRINT "Augen zu und durch..."
   ' GOTO cleanup1
END IF
'
Act&=PEEKL(ioRequest&(kan) + 32)
'
RETURN
'
'
cleanup1:
   REM  *** Close CDTV.Device ***
   CALL CloseDevice(ioRequest&(0))
   CALL CloseDevice(ioRequest&(1))
'
cleanup2:
   REM  *** DeleteExtIO ***
   FOR ip=0 TO 1
     POKE (ioRequest&(ip) +  8), &HFF
     POKEL(ioRequest&(ip) + 20), -1
     POKEL(ioRequest&(ip) + 24), -1
     NEXT
'
cleanup3:
   REM  *** DeletePort ***
   FOR ip=0 TO 1
     CALL RemPort(msgPort&(ip))
     POKE (msgPort&(ip) + 8), &HFF
     POKEL(msgPort&(ip) + 20), -1
     CALL FreeSignal(sigBit%(ip))
     NEXT
   CALL FreeMem(msgPort&,RAMsize&)
'
cleanup4:
   LIBRARY CLOSE
'
SYSTEM
' END
```

Vorstellung

Willkommen im Amiga-Club

In dieser Rubrik werden wir Sie in Zukunft über die Aktivitäten von Computer-Clubs unterrichten, die sich unter anderem (oder ausschließlich) mit dem Amiga beschäftigen. Darüber hinaus soll diese Seite das Forum für Computer-Clubs werden, die Veranstaltungen planen oder ausführen und dies einer breiteren Öffentlichkeit bekanntmachen wollen.

In der heutigen Ausgabe beschränken wir uns auf das Vorstellen einiger Clubs, die uns Informationen zur Verfügung gestellt haben.
Beginnen wir mit dem Computer-Club "PD 2000", dessen Leiter uns die folgenden Zeilen zugesandt hat:

Name und Anschrift:
"PD 2000"
Dirk Timmermann
Postfach 1209
W-2833 Harpstedt
Computertypen: *alle Amiga*
Leistungen: *großer PD-Pool mit neuester Software, zweimonatliches Club-Magazin, super Software- und Hardware-Angebote, Händlerrabatte, großer Compu-Treff im März, eigene PD-Serie (PD 2000), Einsteigerhilfen, Kontakte knüpfen, intensiver Informationsaustausch, Gebrauchtwarenmarkt*
Beiträge: *keine*
Schwerpunkte: *andere User von der Qualität der PD überzeugen und sie somit von Raubkopien abbringen*
Gründung und Mitglieder: *1990/91, ca. 100*
Bemerkungen für Interessierte: *frankierten Rückumschlag beilegen, bei PD-Wunsch Disketten mit Tausch-Software (PD) oder 1 DM je Diskette beilegen*

Zwar etwas knapp beschrieben, aber sicher interessant, der Club "PD 2000". Wir werden bestimmt einmal ausführlicher über ihn berichten können.
Der zweite Club, den wir in dieser Ausgabe vorstellen wollen, ist der ACC Amiga Club Coburg e.V. Aber lassen wir den Schreiber des Briefes über den Club, Frank Eisenwiener, selbst zu Wort kommen:

Wir sind ein im Jahre 1987 gegründeter Verein mit derzeit 65 Mitgliedern unterschiedlichen Alters und unterschiedlicher Berufsgruppen, aber mit gleichem Interessengebiet – dem Amiga.
Der Mitgliedsbeitrag beträgt 60,– DM jährlich, wobei unsere zweimonatlich erscheinende Club-Zeitung auf Diskette (deren Inhalt aus Problemlösungen, eigens erstellten Sound-Kreationen und Animationen, Club-Neuheiten und von Mitgliedern geschriebenen Programmen besteht) im Beitrag enthalten ist.
Außerdem ist ein eigener Club-Rechner vorhanden, der zu Schulungszwecken und für Einsteigerkurse verwendet wird. Zusätzlich dazu haben wir PD-Abos von den bekannten Serien "Fish" und "Time", und bei einigen Händlern bekommen wir bei Sammelbestellung Rabatt.
Jeden Samstag findet unser Treffen mit Erfahrungsaustausch, PD-Service und Problemlösungen statt. Einmal jährlich organisieren wir eine Veranstaltung, die "Coburger Amiga-Tage", bei der die Mitglieder ihr Können auf ihren Amigas der Öffentlichkeit vorstellen. Regelmäßige Messebesuche des Vereins sind auch schon zur Selbstverständlichkeit geworden.
Hauptziel des Clubs ist es auch, den unerfahrenen und jugendlichen Amiga-Besitzern den Umgang mit dem Rechner zu erleichtern.
Die Kontaktadresse lautet:
Amiga Club Coburg e. V.
Postfach 1112
W-8630 Coburg
Die Adresse des ersten Vorsitzenden:
Frank Eisenwiener
Neustadter Str. 11
W-8630 Coburg

Der nächste Club heißt "Extrem". Wollen wir uns mal anhören, was er an "Extremen" zu bieten hat:

Extrem – Der bessere Club! Amiga, C-64, C-128, PC, Atari ST, Game Boy, Nintendo, Game Gear, Sega Mega Drive und Sega Master System
Der Club bietet:
– verbilligten Einkauf von Leerdisketten, Etiketten, Farbbändern, Endlospapier, Reinigungs-Sets, Joysticks usw.
– Jahreswettbewerb mit einem Amiga 500 Plus als Hauptgewinn
– 30seitige, gedruckte Club-Zeitschrift (erscheint alle zwei Monate) mit kostenlosen Kleinanzeigen, Infos, Tests etc.
– Tips-&-Tricks-Archiv
– Programmierhilfen für alle gängigen Programmiersprachen
– Einsteigerhilfen
– monatliche Zusendung der neuesten FD-Software auf einer Diskette
– gemeinsame Software-Entwicklung (zum Beispiel Spiele, Anwendungen etc.)
– Erfahrungsaustausch auf den Clubtreffen (Termine werden frühzeitig bekanntgegeben)
– Hotlines (von 15 bis 21 Uhr)
– Hilfe bei Hardware-Problemen
Der monatliche Beitrag beträgt 6 DM, wer jährlich zahlt, bekommt 12 DM Nachlaß, braucht also nur 60 DM zu zahlen. Die Adresse für Informationen lautet wie folgt:
Stefan Hüls
Dinxperloer Str. 87
W-4290 Bocholt

Der letzte Club, der in dieser AMIGA-DOS-Ausgabe vorgestellt wird, ist der "Power-World-Club", ebenfalls in Bocholt ansässig.

Allgemeines: *Bestehen seit Mitte 1991, Systeme Amiga und C64*
Bedingungen: *Club-Beitrag nach Zahlungswunsch zwischen 5 und 20 DM*
Mitgliederzahl: *zur Zeit 66*
Aktionen/Leistungen: *Amiga/C64-PD-Archiv, monatliche Software-Lieferung von zwei bis zehn Disketten und News, Copy-Parties (nur PD), eigene PD-Serien, Digi-Service, Spiele (PD), Wettbewerbe und Software-Tauschbörsen*
Besonderheiten: *sämtliche Software direkt aus der Szene, maximal eine Woche alt; Infopaket mit Testdiskette gegen 2,50 DM Rückporto; Mailbox ab Mitte 1992*
Kontaktadresse:
Power-World-Club
J.D. Mallander
Knufstr. 28
W-4290 Bocholt

So, das war es für heute.

Bis in einem Monat,
Ihre AMIGA-DOS-Redaktion

Kurzer Hinweis für Clubs, die bei uns vorgestellt werden möchten:
Uns reicht ein Brief, in dem "technische" Informationen über den Club enthalten sind: Ausstattung mit Rechnern (müssen nicht nur Amigas sein, sie sollten aber im Vordergrund stehen), Mitgliederzahl, Club-Leistungen und Mitgliedsbeiträge sowie Informationen über Ziele und Veranstaltungen.
Was wir nicht wollen, sind Hinweise über illegales Kopieren von kommerzieller Software. Clubs, die sich für die Verbreitung von FD-(PD-) Software einsetzen, sind uns dagegen sehr willkommen.

Die Briefe senden Sie bitte an:
DMV-Verlag
Redaktion AMIGA DOS
Postfach 250
W-3440 Eschwege
Kennwort "Amiga-Club"

Geordnete Tips zu PD-Programmen

Leitfaden durch den PD-Dschungel

Wer heute den Durchblick im PD-Wirrwarr behalten hat, der kann sicherlich als Kandidat für "Wetten, daß ..." gelten. Doch es gibt auch noch einen anderen Weg.



Monat für Monat werden die neuesten Programme aus den vielen Ländern und Orten auf Disketten veröffentlicht. Die meisten Programme kommen dann als PD (Public Domain) oder als FD (Freely Distributible) in die verschiedensten Serien auf dem Markt. Doch wer soll bei dieser Vielzahl von Disketten und Programmen den Überblick behalten? Wohl keiner.

Darum hat sich der Scheider-Verlag Gedanken gemacht und eine Art Leitfaden entwickelt, der sich mit Programmen im PD-, Shareware- und Freeware-Bereich auseinandersetzt.

Das Prinzip, nach dem hier vorgegangen wird, ist recht einfach erklärt: Programme, die aus dem riesigen Pool der PD-, FD-, Freeware- und Shareware-Serien stammen, werden verglichen. Die Programme, die mit besonderen Merkmalen aufwarten, werden in einer Liste aufgenommen. Diese Programme werden dann ihrem jeweiligen Anwendungsbereich zugeordnet, zum Beispiel CAD dem Bereich Grafik. Anschließend werden die Programme mit einer ausführlichen Beschreibung über Funktion, Zweck, Besonderheiten und Installation versehen. Natürlich werden auch der Autor und dessen Vorgaben über Vertrieb mitgegeben. Um das Ganze übersichtlicher zu machen, werden die Beschreibungen in einem DIN-A5-Ordner ihrem jeweiligen Bereich zugeordnet und abgeheftet.

Die Serien

Werden im ersten Band noch die Grundsätze des Gedankens "Public Domain" beschrieben und die Begriffe wie Shareware und Freeware erläutert, so wird einem nach weiterem Lesen bald klar, nach welchen Kriterien die Programme ausgewählt wurden. Durch die vielfältigen Anwendungen wurden Gliederungen der Sparten eingeführt. Aufgeführt werden hier: kommerzielle Software (Shareware), Grafik und Animation, Sound und Sprache, Spiele, Utilities, Programmiersprachen, Programme für Haushalt, Schule, Technik und Wissenschaft. Nach und nach werden neue Anleitungen herausgebracht, die in dem Ordner untergebracht werden.

Da der Spielebereich natürlich auch groß ist, gibt es für den Spielefreund eine Übersicht unterhaltsamer Spiele, die aus Denk-, Action-, Strategie- und Geschicklichkeitsspielen besteht.

Spiele-Spezial 2

Der Ringbuchordner Band 2 enthält außer den gelochten und locker geschriebenen Seiten auch noch vier Disketten, die mit den unterschiedlichsten Spielen aufwarten.

Diskette 1 enthält folgende Spiele.

China Challenge
Dies ist eine Variante des bekannten Logikspiels »Shanghai«.

Cross
Verstecken Sie eine von Ihnen bestimmte Anzahl von Wörtern in einem vordefinierbaren Koordinatensystem. Für diejenigen, die einen Drucker besitzen, ist die Möglichkeit des Ausdruckens gegeben.

Dominos
Wer kennt es nicht, das alte Steinspiel, das jetzt in einer Computerumsetzung vorliegt? Drei Schwierigkeitsgrade sind wählbar, die gegen den Computer zu spielen sind.

FlipIT
Ein Denkspiel, das auch unter dem Namen "Reversi" bekannt sein dürfte.

Globulus
Ein kommerzielles Spiel als spielbare Demo. Führen Sie "Globy" durch das dreidimensionale Spielfeld ans Ziel.

Logic
Ein Spiel, das die logischen Fähigkeiten jedes Spielers fordert. Neun schwarze Quadrate sind so anzuklicken, daß acht Felder weiß umrandet sind und ein Feld in der Mitte schwarz bleibt.

Die zweite Diskette enthält nur ein Spiel, das es aber in sich hat:

Battle Force
Durch den Umfang und die Komplexität des Simulationsspiels wurde auch die Beschreibung dementsprechend ausgelegt. Fast die Hälfte der Beschreibung befaßt sich in allen Einzelheiten mit dem Umgang mit dem Programm, und man erfährt viele notwendige Tips und Tricks.

Die dritte Diskette enthält:

Eishockey
Managen Sie in einer Simulation Ihre eigene Eishockey-Mannschaft.



Mines
Finden Sie Bomben anhand von Zahlen.

MiniBlast
Ein hübsches kleines Ballerspiel für zwischendurch.

Patience
Das bekannte Kartenspiel in einer schönen Variante.

Pythagoras
Eine Wirtschaftssimulation in der Zeit des alten Rom. Mit schöner Grafik.

Raps
Versuchen Sie, mit vier Spielsteinen die gegnerische Linie zu erreichen.

StepZ
Auch hier müssen Sie die gegnerische Linie erreichen. Doch Vorsicht! Die Gegner sind Richtungspfeile, die versuchen, Sie aufzuhalten.

Yawn!
Ein Verschiebefix mit Zahlen. Unterschiedliche Zahlendarstellung und Größen sind einstellbar.

Auf der vierten Diskette finden sich weitere Spiele.

Alle in diesem Artikel beschriebenen Programme werden so ausreichend besprochen, daß auch englische Programme verstanden werden, was vor allem dem Benutzer zugute kommt, der nicht der englischen Sprache mächtig ist. Am Schluß findet sich noch eine kurze Übersicht der zur Zeit aktuellen Spiele. ❑

W. Meiß (gg)

Info
Schneider-Verlag, Am Weinberg 46, 8301 Arth, Tel.: 08704-1597
Preis: 98,– DM
Disketten: 39,– DM
Spiele-Spezial 2 (inkl. Disk): 59,– DM

Elektronisches Gestalten

Der Titel könnte irritieren. "Elektronisches Gestalten in Kunst und Design" ist kein Buch, das zum Desktop-Layout anleitet, Mal- und 3D-Programme vorstellt oder den idealen Bildaufbau diskutiert. Der Untertitel "Richtungen – Institutionen – Begriffe" weist auf etwas Umfassenderes hin. Elektronisches Gestalten sei viel mehr als nur die enge Anwendung am Bildschirm, betont Jürgen Claus in seinem neuesten Buch. Autor Claus, Medienkünstler und Professor der Kölner Medienhochschule, kennt sein Thema aus Praxis und Theorie. Mit "fraktalem Humor" – so er selbst über sein Bestreben, als Insider Distanz zu üben – setzt er Beschaffenheit und Bedeutung der neuen Medien in den Kontext. Auf knapp 190 Seiten bringt er



einen Überblick über das Spektrum der Medien, berichtet aus europäischen Zentren der Technokultur und erklärt eingehend Schlagworte aus der Szene. In der Mitte zeigt ein 16seitiger Farbteil Performances, Installationen und Grafiken. Auf den letzten 15 Seiten stehen umfangreiche Personen- und Sachregister und Literaturhinweise, die nach Themen geordnet sind. Für Leser mit der schrecklichen Angewohnheit, Sachbücher auf den letzten Seiten anzufangen, ist bereits diese Liste weiterführender Literatur und Artikel eine wahre Goldgrube. Freilich ist sie kein Grund, nicht doch noch am Anfang mit dem Lesen zu beginnen.

Abschließend, im "Lexikon aktueller Begriffe", liefert der Autor Definitionen, historischen Hintergrund und kreative Einsatzmöglichkeiten von so unterschiedlichen Technologien und Philosophien wie Computeranimation, Hypermedia, Robotik und Virtueller Realität. Schade nur, daß der Bereich der kleinen kreativen Computer im Mediengeschehen ein Stiefkinddasein führt. Gerade in der Ausbildung haben sich doch erschwingliche, leistungsfähige Systeme wie der Amiga bereits etabliert.

"Elektronisches Gestalten in Kunst und Design" ist trotz der gebotenen Informationsfülle gut zu lesen. Als Außenstehender fehlt einem ab und zu der Durchblick, wenn Claus kunst- und medienhistorisches Fachwissen voraussetzt. Aber dann fängt man eben wieder am Schluß des Buches an und sucht nach ganz themenspezifischen Büchern. ❑

Ute Bahn (gg)

Titel: Elektronisches Gestalten in Kunst und Design
Autor: Jürgen Claus
Verlag: Rowohlt-Verlag, 1991
Preis: 16,80 DM
ISBN: 3-499-18194-0

Wissen

Das AMIGA-DOS-Computer-Lexikon

Rund um den Amiga tauchen immer wieder Spezialbegriffe auf, die einer genaueren Erklärung bedürfen. Mit diesem Lexikon wollen wir versuchen, Ihnen ein wenig beim Erlernen und Begreifen behilflich zu sein.



A

Agnus: Einer der → Customchips im Amiga. Übernimmt Speicherverwaltung, → DMA und Coprozessorverwaltung. Agnus gibt es in verschiedenen Versionen: Agnus (im A1000), Fat Agnus (frühere A500/A2000) und Big Agnus (neuere A500/2000). Im A3000 findet eine erweiterte Version des Big Agnus Anwendung.

Amiga: spanisch "Freundin".

AmigaOS: "Amiga Operation System", das → Betriebssystem des Amiga.

ANSI-Terminal: Standard für Bildschirmdarstellung. Umfaßt Steuerzeichen für die Art der Textdarstellung (Fett, Unterstrichen, Kursiv, Farben) und für den Zeichensatz. Die Amiga-Textausgabe entspricht größtenteils dem ANSI-Standard.

Autoconfig: Verfahren, mit dem sich Erweiterungskarten im Amiga-System selbsttätig anmelden und Speicherbereiche zugewiesen bekommen (→ Zorro). Damit sind Kodierungsjumper wie im PC-Bereich überflüssig.

B

Betriebssystem: Programm, das beim Starten des Computers abgearbeitet wird. Das Betriebssystem stellt zum Beispiel Funktionen für Ein- und Ausgabe auf Bildschirm oder Diskettenlaufwerke zur Verfügung. Ohne Betriebssystem könnte ein Computer nicht starten.

Bit: Kleinste Informationseinheit im Computerbereich. Kann entweder die Werte 0 oder 1 annehmen ("falsch" oder "wahr").

Blitter: engl.: "Bit Image Transferer". Coprozessor für das schnelle Verknüpfen und Verschieben von Daten. Hauptsächlich für Grafikanwendungen (Verschieben von grafischen Objekten). Kann auch Linien ziehen oder Flächen füllen. Ist im → Agnus integriert.

Byte: Informationseinheit, bestehend aus 8 Bits. Kann Werte von 0 bis 255 annehmen. Auch Maßeinheit für die Größe von Speicher: 1 KByte = 1024 (!) Byte, 1 MByte = 1024 KByte. Die oftmals gebrauchten Bezeichnungen "Kilobyte" und "Megabyte" (Kilo = Faktor 1000, Mega = Faktor 1000000) sind daher verkehrt.

C

Cache: Grundsätzlich: schneller Zwischenspeicher zur Ablage von Daten von langsamen Quellen. Zum Beispiel können Daten von einer Festplatte im RAM gehalten werden, damit sie bei Abruf sofort zur Verfügung stehen. Prozessoren ab 68020 aufwärts besitzen einen Cache, der Daten aus dem vergleichsweise langsamen RAM zwischenspeichert und der Prozessor daraus ohne zusätzlichen RAM-Zugriff lesen und schreiben kann.

Centronics: Standard für eine → parallele Druckerschnittstelle. Die parallele Schnittstelle im Amiga ist Centronics-kompatibel, kann aber im Gegensatz dazu auch für Eingabezwecke verwendet werden.

Chip-Memory: Teil des RAM, auf den die → Customchips zugreifen können. Grafik- und Sound-Daten müssen unbedingt in diesem Speicherbereich liegen. Maximale Größe ist abhängig vom Agnus-Typ: normaler Agnus und Fat Agnus maximal 512 KByte, Big Agnus entweder 1 MByte oder 2 MByte. → Fast-Memory, → Ranger-Memory. Das Chip-Memory wird während des Reset-Vorgangs automatisch eingebunden.

CLI: "Command Line Interface". Textorientierte Benutzeroberfläche des AmigaOS. Inzwischen durch die komfortablere → Shell abgelöst.

CIA: "Complex Interface Adaptor". Zwei dieser Chips (8520) steuern im Amiga Tastatur, parallele Schnittstelle und verschiedene andere Ein- und Ausgabeleitungen (Power-LED, Maus- oder Joysticktasten, Teile der seriellen Schnittstelle). Die CIAs sind sehr empfindliche gegen statische Elektrizität und können durch unsachgemäße Handhabung der externen Anschlüsse leicht zerstört werden; Hauptfehlerursache bei defekten Amigas.

Clipboard: Englisch für "Klemmbrett". Vorrichtung, um Daten kurzfristig abzulegen. Wird zum Beispiel von Texteditoren benutzt, um "ausgeschnittene" Textteile zwischenzuspeichern. Unter OS 2.0 kann Text aus beliebigen Textfenstern ausgeschnitten und importiert werden.

Controller: In der Regel die Steuerlogik zur Ansteuerung von → ST-506-Festplatten oder Diskettenlaufwerken (SCSI-Festplatten haben den Controller integriert). Der Controller übernimmt die Verwaltung und Aufbereitung der Daten für die Speicherung auf dem Datenträger (→ Datenaufzeichnung). Aus diesem Grund ist die nutzbare Kapazität vom Datenaufzeichnungsverfahren des Controllers abhängig; man unterscheidet dabei in der Hauptsache → MFM und → RLL.

Copper: Einer der Coprozessoren im Amiga. Kann abhängig von der Rasterstrahlposition Farbwerte, Auflösung und andere Werte der Bildschirmausgabe beeinflussen. Ist im → Agnus integriert.

CPU: "Central Processing Unit" → Prozessor.

Customchip: Chip, der für eine spezielle Anwendung entwickelt wurde. Das Amiga-System verdankt seine überragende Leistungsfähigkeit den Customchips → Agnus, → Denise, → Gary und → Paula

D

Datenaufzeichnung: Auf Festplatten und Disketten können Daten nicht im Rohzustand geschrieben werden. Zu viele Bits mit gleichem Wert würden in den Gleichlaufschwankungen des Antriebs untergehen und wären nicht mehr unterscheid-

bar, so daß Lesefehler aufträten. Daher werden die Daten vor der Aufzeichnung kodiert, um Häufung gleicher Bits zu vermeiden; dadurch wird die nutzbare Kapazität des Mediums verringert. → MFM, → RLL, → GCR.

Denise: Einer der → Customchips im Amiga, verantwortlich für die Bildschirmausgabe. Neue Version mit neuen Bildschirmmodi (oft HR-Denise genannt) findet sich im A3000 und A500 Plus.

Device: Die Steuersoftware für Hardware. Alle Devices werden über den gleichen Mechanismus angesteuert. Beispiele: »serial.device«, »trackdisk.device«. Einige Device-Treiber sind im ROM enthalten, andere liegen auf der Workbench-Disk im Ordner »DEVS:«.

DOS: "Disk Operating System"; der Teil des → Betriebssystems, der für die Verwaltung von Massenspeichern (Dateien) zuständig ist. Das DOS wird zuletzt im System gestartet und nimmt Eingaben vom Benutzer entgegen. AmigaDOS besteht aus mehreren Teilen: Der »dos.library« im ROM, diversen integrierten → Handlern im ROM und den Shell-Befehlen auf der → Workbench-Diskette.

DMA: "Direct Memory Access", direkter Speicherzugriff. Methode, mit der beliebige Teile der Rechnerhardware (auch externe Erweiterungen) auf das → RAM des Rechners zugreifen können, ohne daß der Hauptprozessor benötigt wird. Im Amiga ist DMA durch die → Customchips nur im → Chip-Memory möglich. Bus-DMA (→ Zorro) geht in jedem Speicherbereich.

E

ECS: "Enhanced Chip System". Bezeichnung für die neuen Versionen von → Agnus und → Denise (Big Agnus und HR-Denise).

Exec: "Executive". Der Kern des → AmigaOS, verwaltet das Multitasking-System.

F

Fast-Memory: Speicherbereich, der nicht von den → Customchips ansprechbar ist. Da keine Störung durch → DMA-Zugriffe der Customchips, laufen Programme im Fast-Memory etwas schneller. → Ranger-Memory wird oft fälschlicherweise als Fast-Memory bezeichnet. Fast-Memory wird, außer beim A3000, nicht vom Betriebssystem eingebunden, sondern muß sich selbst per → Autoconfig anmelden. Eine besondere Art des Fast-Memory ist das sogenannte 32-Bit-RAM auf verschiedenen Turbokarten und im A3000. Auf dieses RAM kann der Prozessor (ab 68020) direkt 32 Bit breit und daher wesentlich schneller zugreifen (→ Turbokarten).

FD: "Freely Distributable"; Software, die mit Einschränkungen frei verteilt werden darf. Oberbegriff für → Public Domain, → Freeware und → Shareware.

FD-Files: Dateien, die die Aufrufe der Funktionen einer Library beschreiben. Anhand dieser Dateien kann jede Programmiersprache beliebige Libraries benutzen.

Freeware: Vom Autor zum Vertrieb freigegebene Software. Der Autor behält sich alle Rechte vor und kann beliebige Beschränkungen des Vertriebs zur Auflage machen (zum Beispiel, daß die Software nicht von sogenannten PD-Händlern verkauft werden darf). → Public Domain, → Shareware.

G

Gary: → Customchip im Amiga. Unterstützt die Speicherverwaltung.

GCR: "Group Code Recording". → Datenaufzeichnungsverfahren, bei dem die aufzuzeichnenden Daten gruppenweise kodiert und durch entschärfte, aber längere Codes ersetzt werden. Im Gegensatz zu → MFM wird die Nutzkapazität nur um etwa 25 Prozent verringert, dafür ist das Kodierverfahren sehr rechenaufwendig. Wird heute kaum noch eingesetzt.

H

Handler: Eine DOS-Erweiterung, die den Zugriff auf bestimmte Daten wie auf eine Datei erlaubt. Beispiele sind »SPEAK:«, das wie eine DOS-Datei funktioniert und beim Beschreiben den Text über die Amiga-Sprachausgabe ausgibt. Einige Handler sind im ROM enthalten ("CON:", "SER:", "PAR:", "PRT:"), andere liegen auf der Workbench-Disk im Ordner »L:«.

I

IFF: "Interchange File Format". Genormtes Format für die Ablage von Daten. So können Bilddateien aus einem Programm problemlos in ein anderes eingeladen werden. IFF-Formate existieren für diverse Anwendungszwecke; am häufigsten trifft man ILBM für Bilddaten an. Weitere Standards sind zum Beispiel ANIM für Animationen auf ILBM-Basis, 8SVX für gesamplete Sound-Daten, FTXT für Text.

Intuition: Die grafische Benutzeroberfläche des → AmigaOS. Verwaltet Windows und Screens und regelt die Eingabe.

Interface: Englisch für → Schnittstelle.

Interlace: Verfahren zur Bildschirmdarstellung, bei dem abwechselnd die geraden und ungeraden Zeilen eines Bildes dargestellt werden. Ergibt höhere Auflösung ohne technischen Mehraufwand, dafür ist das Bild, gerade bei sehr kontrastreichen Darstellungen, ausgesprochen "verflackert". Wird vom Fernsehen verwendet und fällt dort wegen der schlechten Qualität der Fernsehgeräte (Verwaschung) kaum auf.

K

Kickstart: Bezeichnung für die Startdiskette, von der beim A1000 und beim A3000 das → Betriebssystem geladen wird. Auch: Bezeichnung für das Betriebssystem-ROM im A500 oder A2000 ("Kickstart-ROM") und allgemein für das Amiga-Betriebssystem.

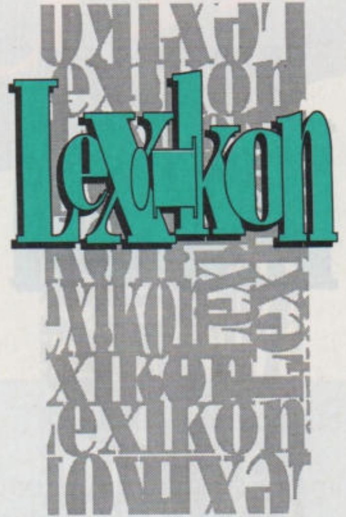
L

Library: Englisch für "Bibliothek". Funktionssammlungen mit bestimmten Aufgaben. Das Amiga-Betriebssystem besteht in der Hauptsache aus verschiedenen Libraries ("exec.library", "dos.library", "intuition.library"). Die Libraries sind entweder im ROM enthalten oder liegen auf der Workbench-Disk im Ordner »LIBS:«. Libraries haben stets gleichlautende Aufrufe und können so von allen Programmiersprachen benutzt werden (→ FD-Files).

M

MFM: "Modified Frequency Modulation". → Datenaufzeichnungsverfahren, bei dem jedes Bit ein zusätzliches Taktbit bekommt. Die Nutzkapazität sinkt um 50 Prozent, dafür ist das Verfahren sehr schnell. Amiga-Disketten werden mit MFM kodiert (die Kodierung übernimmt der Blitter).

Multitasking: Möglichkeit des → Betriebssystems, mehrere



Lexikon

Programme scheinbar gleichzeitig ablaufen zu lassen. Verlangt bestimmte Programmiertechniken, um die Rechenzeit vernünftig einzuteilen. Im AmigaOS verbraucht ein Programm, das auf bestimmte Ereignisse (zum Beispiel Eingabe) wartet, keine Rechenzeit.

P

parallel: Bei → Schnittstellen: Daten werden durch mehrere Leitungen parallel übertragen. Oft werden 8 Leitungen verwendet (→ Byte).

Paula: → Customchip im Amiga. Verwaltet Audio-Ausgabe, Disketten-Controller und serielle Schnittstelle.

Prozessor: Kernstück eines Computers. Steuert die übrige Hardware des Systems und arbeitet Programme ab. Manche Rechner haben mehrere Prozessoren: Einen Hauptprozessor für die Programmabarbeitung und zusätzliche Coprozessoren für bestimmte Aufgaben (besonders Grafikbearbeitung). Im Amiga finden Prozessoren der 68000-Familie Anwendung, die sich durch komfortablen Befehlssatz und hohe Leistungsfähigkeit auszeichnen. Die Prozessoren sind uneingeschränkt aufwärtskompatibel. Der 68000 im A1000, A500 oder A2000 kann problemlos gegen einen 68010 ausgetauscht werden, wobei aber der Geschwindigkeitsgewinn minimal ist. Ab 68020 ist zusätzliche Hardware nötig (→ Turbokarte). Der A3000 hat standardmäßig einen 68030 eingebaut.

Public Domain: Englisch für "öffentliches Eigentum". Software, bei der der Autor auf alle Urheberrechte verzichtet. PD-Soft darf beliebig kopiert und verwertet werden. PD wird oft fälschlicherweise als Oberbegriff für Public Domain, → Freeware und → Shareware benutzt. Juristisch gesehen ist es Geschäftsschädigung, Shareware als PD anzubieten.

R

RAM: "Random Access Memory", Speicher mit wahlfreiem Zugriff. Der Arbeitsspeicher des Rechners.

Ranger-Memory: Speicherbereich, der während des Resets vom Betriebssystem eingebunden wird. Zwar kein → Chip-Memory, wird aber von → DMA-Zugriffen ebenso gebremst. Interne Erweiterungen im A500 sind praktisch immer Ranger-Memory.

RLL: "Run-Length-Limited". → Datenaufzeichnungsverfahren ähnlich → GCR, aber effektiver und mit integrierter Fehlerkorrektur. RLL verlangt Festplattensysteme hoher Qualität.

ROM: "Read Only Memory", Nur-Lese-Speicher. Theoretisch wie → RAM, kann aber nur gelesen werden. Enthält in der Regel Teile des → Betriebssystems.

RS-232: Standard für eine serielle → Schnittstelle. Die im Amiga eingebaute serielle Schnittstelle entspricht im wesentlichen diesem Standard.

S

Schnittstelle: Allgemeine Bezeichnung für Verbindungen zwischen zwei Rechnern oder zwischen Rechnern und → Peripherie. Damit eine Verbindung klappt, muß die Schnittstelle an beiden Seiten von der gleichen Art sein. Oft werden serielle (→ RS-232) und parallele (→ Centronics) Schnittstellen unterschieden.

Screen: Englisch für "Bildschirm". Im Gegensatz zu anderen Benutzeroberflächen erlaubt → Intuition mehrere Bildschirme, die vertikal (ab OS 2.0 auch horizontal) verschoben werden können. Unterschiedliche Screens können unterschiedliche Modi und Farben haben.

SCSI: "Small Computer System Interface". Parallele Schnittstelle für Datenübertragungen bei hoher Geschwindigkeit. Verlangt von den angeschlossenen Systemen viel "Eigenintelligenz", erlaubt aber so, völlig unterschiedliche Geräte (Festplatten, Streamer) auf gleiche Art und Weise anzusteuern. Eine SCSI-Verbindung kann aus maximal acht Geräten bestehen. Für die Rechnerseite wird ein sogenannter SCSI-Hostadapter benötigt, fälschlicherweise oft SCSI-Controller genannt. SCSI-Festplatten haben den Controller integriert, im Gegensatz zu → ST-506-Platten.

seriell: Bei → Schnittstellen: Daten werden über eine einzige Leitung bitweise übertragen (vergleiche auch → parallel).

Shell: Textgesteuerte Benutzeroberfläche des AmigaOS. Bis Kickstart 1.2 auch → CLI genannt. Siehe → Workbench.

ST-506: Schnittstelle für Festplattensysteme. Die Platte wird dabei direkt wie ein Diskettenlaufwerk angesteuert. Auswertung und Aufbereitung der Daten erfolgt durch einen → Controller.

Shareware: Software, die unter bestimmten Auflagen kopiert werden darf. Bei Gefallen muß die Software bezahlt werden. Shareware ist weder → Public Domain noch → Freeware, sondern eine besondere Vermarktungsform. Mischformen zwischen Freeware und Shareware, bei der der Autor eine Bezahlung erbittet, aber nicht unbedingt verlangt, werden in letzter Zeit auch als Giftware bezeichnet (englisch "Gift" = Geschenk).

T

Turbokarten: Erweiterungen mit neuem, schnellerem Prozessor, meistens 68020 oder 68030. Diese Prozessoren werden schneller getaktet als der 68000 im Standard-Amiga (7,1 MHz), sind intern optimiert, besitzen → Cache und arbeiten mit echtem 32-Bit-Datenbus. Turbokarten sind nur dann sinnvoll, wenn sie auch 32-Bit-RAM (→ Fast-Memory) enthalten, ansonsten wird ein Großteil des Geschwindigkeitsgewinns vergeben. Das Amiga-Betriebssystem unterstützt die gesamte 680x0-Familie, so daß keine Probleme auftreten (außer durch schlecht programmierte Software, in der Hauptsache Spiele).

W

Window: Englisch für "Fenster". Abgegrenzter Bereich für Ein- und Ausgabe. Windows werden von → Intuition verwaltet.

Workbench: a) Bezeichnung für die Start-Diskette des AmigaOS; b) Oberfläche zur Bearbeitung von Dateien unter Benutzung der grafischen Benutzeroberfläche Intuition.

Z

Zorro: Bezeichnung für das Erweiterungssystem (Bussystem) der Amiga-Familie. Zorro-Bus beim A1000, Zorro-II beim A2000 (beim A500 nur ansatzweise) und Zorro-III beim A3000. Zorro-II-Karten können im Zorro-III-Bus verwendet werden. Eingeschränkt ist die Nutzung von Zorro-II-Karten am A500 und A2000 möglich. ❑

Jürgen Stephan (jb)

AMIGA-DOS-Pixelpanorama

Bekannte Namen und Newcomer: Im Pixelpanorama trifft sich die Grafikerelite der 90er Jahre.

Einen unverwechselbar plastischen Malstil beherrscht Udo Drüke aus Norden (im Norden). Bereits zum zweitenmal ist er im Pixelpanorama vertreten, diesmal mit seinem Werk "Ami da Vinci". Kreiert hat er es ausschließlich mit »Deluxe Paint IV«.

Ebenfalls ein alter Bekannter der Amiga-Grafikszene ist Tobias Richter, der bereits auf diversen Amiga-Messen mit seinen Star-Trek-Animationen für Aufsehen sorgte. Inzwischen ist er, wie nicht anders zu erwarten, auf »Reflections 2.0« umgestiegen. Neues Programm, neue Ziele: von Star Trek zum Krieg der Sterne. Aus der Star-Wars-Serie stammt auch das Vorbild für den "Y-Wing", den er gekonnt in Szene setzte.

Das "Kugelparadies" von Christoph Molderings aus Hammersbach zeigt die Verwendung verschiedener Textur-Brushes und Materialien in Raytracing-Programmen. Wir vermuten, daß dieses Bild mit »Reflections« erstellt wurde.

Ein recht explosives Motiv wählte Axel Kessler aus St. Wendel für sein Bild "Diskcrash", das er mit »Deluxe Paint III« schuf. Wir hoffen, er hat einen sehr guten Virenkiller dagegen ...

Wenn Sie sich mit Ihrem Kunstwerk (Raytracing oder Malprogramm) am Pixelpanorama beteiligen wollen, sollten Sie folgendes tun:

1. Ein oder mehrere ansprechende Bilder im IFF-Format (auch 22 Bit bis maximal 1000 mal 1000 Punkte) aus Ihrer Sammlung auf eine Disk kopieren,
2. Ihren Namen und Ihre Adresse auf die Disk schreiben und
3. das Ganze an folgende Adresse schicken:

DMV-Verlag
Redaktion AMIGA DOS
Kennwort: Pixelpanorama
Postfach 250
W-3440 Eschwege

Für uns sind auch Angaben über die verwendeten Programme, Hardware und Rechenzeiten sowie Informationen über den Künstler interessant. Für jedes im Heft abgedruckte Bild gibt es ein Spielepaket. Wir freuen uns auf Ihre Einsendung.

Das war es dann auch schon wieder. Bis zum nächsten Mal!



Bild 1. "Ami da Vinci" von Udo Drüke

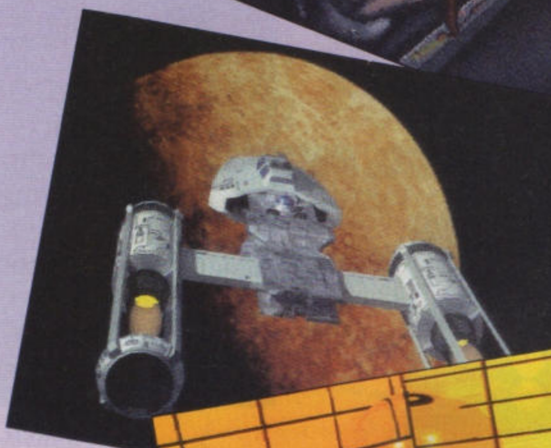


Bild 2. "Y-Wing" von Tobias Richter

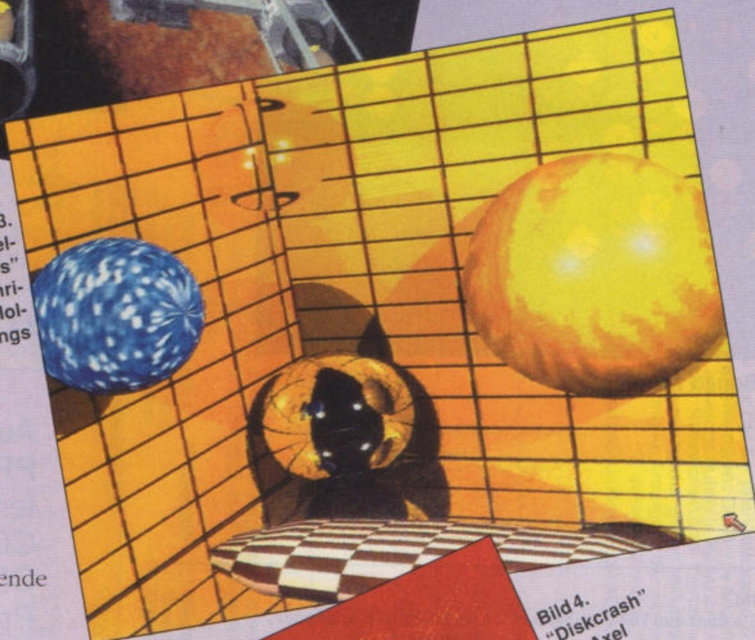


Bild 3. "Kugelparadies" von Christoph Molderings

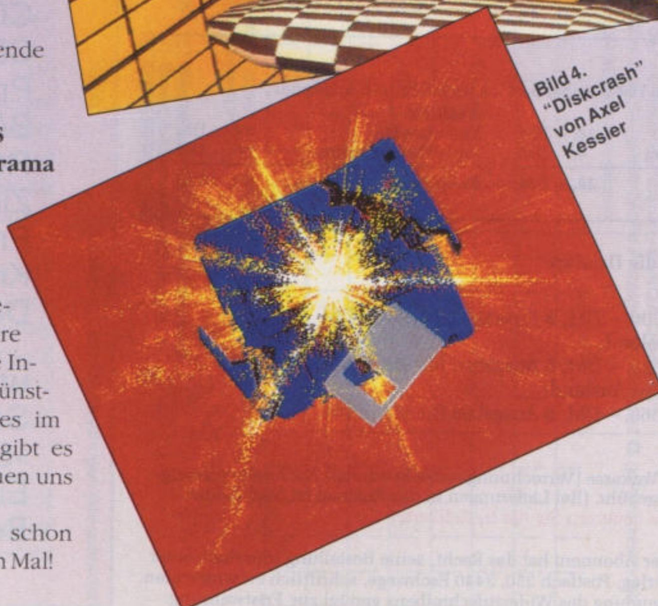


Bild 4. "Diskcrash" von Axel Kessler

DIESMAL:

Workshop

Die Bereiche Tools und Grafik stehen im Vordergrund unserer beiden FD-Workshops. Wir haben die beiden Programme **»GadgetEd«** und **»MandelSquare«** näher unter die Lupe genommen und zeigen Ihnen, wie man mit diesen beiden Programmen arbeitet.

☐ **»GadgetEd«** ist ein Freeware-Tool des Holländers Jan van den Baard. Es präsentiert sich als recht komfortabler Editor, mit dem sich Gadget-Strukturen in C- oder Assembler-Quellcode erstellen lassen.

☐ **»MandelSquare«** von Olaf Barthel ist ein brandneues Freeware-Grafikprogramm, das ausschließlich unter OS 2.0 läuft. Mit »MandelSquare« können nicht nur Fraktale berechnet, sondern auch Fraktalanimationen erstellt werden.

FD-Programme und Tips

☐ Das Anwendungsprogramm **»PicBase«** bringt Ordnung in Ihre Sammlung diverser Grafiken und Bilder.

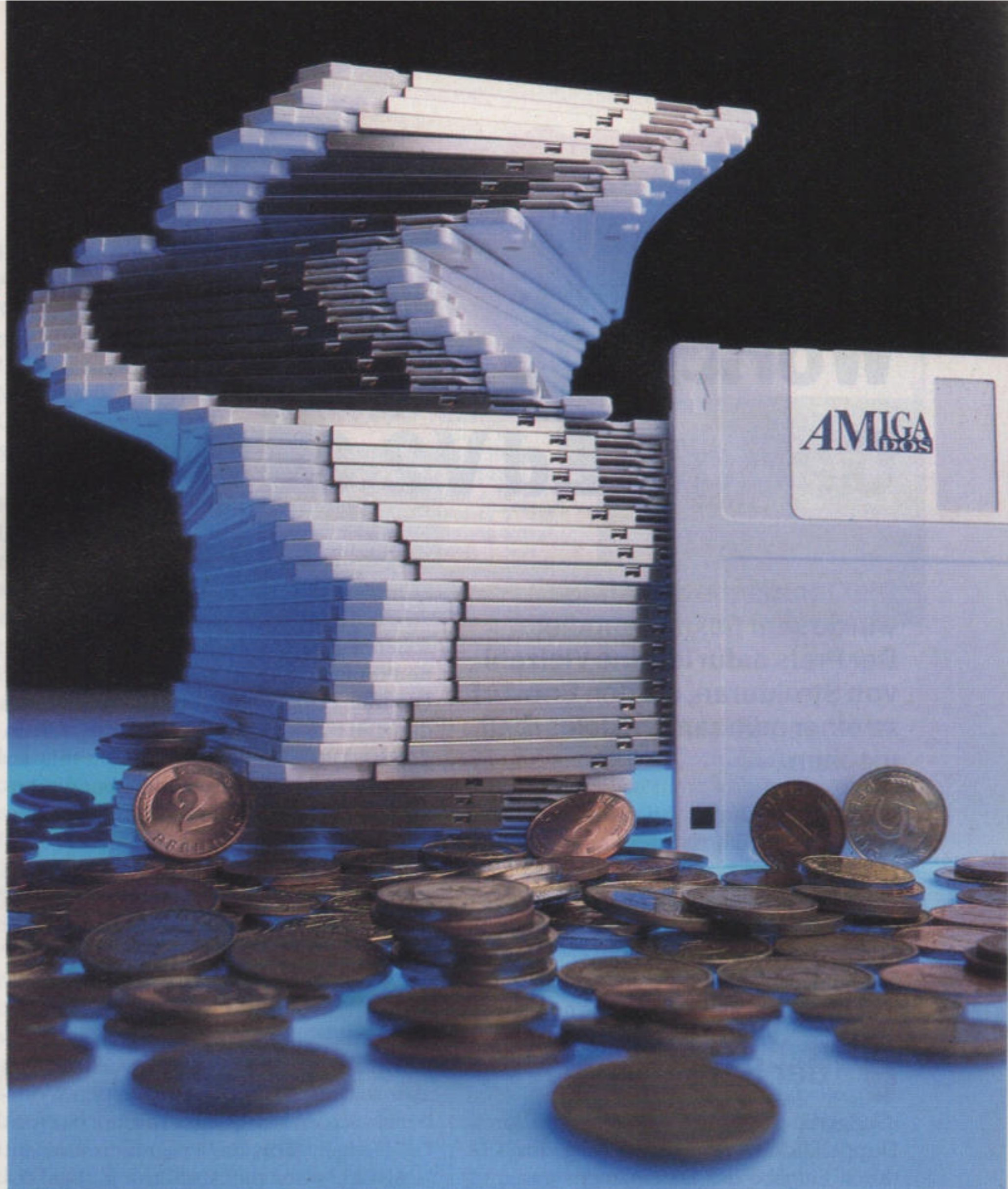
☐ **»DeskBench«** stellt ein probates Mittel dar, wie Sie Ihrem Laserdrucker auf die Sprünge helfen.

☐ **»MachIII«** ist ein recht mächtiges Utility. Wir liefern Ihnen wertvolle Tips dazu.

Das Anwendungsprogramm **»Adresser«** kann für Sie automatisch gespeicherte Telefonnummern wählen.

☐ **»Badger«** zählt ebenfalls zu den Anwendungen und verwaltet Ihre Termine.

☐ **»Typographer«** ermöglicht es, eigene Schriften zu entwerfen und sie dann problemlos mit 24-Nadel-Druckern oder Deskjet-Kompatiblen zu verwenden.



FD-POOL

Serien

☐ Wir stellen Ihnen die Serie "ABCD" vor, die vom Amiga-Basic-Club Deutschland zusammengestellt wurde und sich an alle Amiga-BASIC-Freunde richtet. Ob Einsteigerkurs, Utilities, Dateiverwaltungsprogramme oder Spiele, es ist für jeden sicherlich das Passende dabei.

☐ Die FD-Szene gerät monatlich "in Aufruhr", wenn Neuigkeiten aus Übersee nach Deutschland gelangen. Fred Fish, seines Zeichens Initiator der "AmigaLibDisk"-Reihe, kurz Fish-Disk genannt, bringt diesen Monat die Fis(c)he 571 bis 590 heraus.

Inhalt

Public Domain

Tools

Workshop zu GadgetEd V2.3

Die Oberfläche des Amiga wurde sehr flexibel angelegt. Der Preis dafür ist eine Vielzahl von Strukturen, die den Entwurf zu einer mühsamen Kleinarbeit machen.

Der Holländer Jan van den Baard dachte sich, daß der Amiga für diese Aufgabe besser als der Mensch geeignet ist, und legt mit dem Freeware-Programm »GadgetEd« einen komfortablen Editor vor, der später sowohl C- als auch Assembler-Quellcode erzeugt.

Wie tief soll der Screen sein?

»GadgetEd« kann auf der Workbench durch Doppelklicken des Icons oder im CLI durch Eingabe des Dateinamens gestartet werden. Zusätzlich können Sie im CLI als Argument den Namen eines »GadgetEd«-Files – im folgenden kurz GE-File genannt – anführen.
Diese Datei enthält Daten über eine zuvor mit dem Programm erstellte Oberfläche, die nun bearbeitet wird. Der Editor benötigt zwingend die »tool.library« im Verzeichnis »libs:«, da sonst Ihr Amiga den Programmstart mit einer Guru-Meditation quittiert.
Wurde kein Dateiname angegeben oder das Programm von der Workbench gestartet, erscheint ein Window in der Mitte der Workbench, über das Sie die Screen-Daten für das Window beziehungsweise den Requester festlegen.

Zunächst können Sie wählen, ob ein Window oder ein Requester entworfen werden sollen. Diese Angabe kann aber später ohne Probleme wieder überarbeitet werden. Die fünf Gadgets ("1" bis "5") bestimmen die Tiefe des Screens (englisch: screen depth). Diese Tiefe, die identisch mit der Bitplane-Zahl ist, entscheidet über die Anzahl der Farben. »GadgetEd« öffnet mit Ausnahme der Tiefe "5" einen Hires-Screen, das heißt 640x256 Pixel. Bei "5" hingegen verwendet das Programm einen Lores-Screen mit 32 Farben. Sollen das Fenster beziehungsweise der Requester später auf der Workbench erscheinen, muß das Gadget "Workbench" aktiviert sein.
»GadgetEd« öffnet dann zwar auch einen eigenen Screen, der erzeugte Quellcode ist aber statt auf einen Custom- auf den Workbench-Screen ausgelegt. Klicken Sie das OK-Gadget, fährt das Programm fort, mit "CANCEL" wird die Ausführung abgebrochen.
Sollen das Window beziehungsweise der Requester auf einem Lores-Screen mit weniger als fünf Bitplanes erscheinen, so muß ein Trick angewandt werden. Sie öffnen einen Screen mit der Tiefe "5", verzichten jedoch auf die zusätzlichen Farben. Später gilt es, die eigentlich gewünschte Screen-Tiefe in das Feld "Depth" der New-Screen-Struktur (ist die Struktur, die bei jedem neuen Screen benötigt wird) im Quellcode zu ändern.

Tip für Programmierer

Alternativ kann aber auch die Version 2.3 des »GadgetEd« auf Lores-Auflösung gepatcht werden. Da das Programm in gepackter Form vorliegt, muß es zunächst mit dem »PackerPacker 3.0« als ursprüngliches, nicht gepacktes File (über 90 KByte) gespeichert werden. Ändern Sie anschließend mit einem Datei-Editor im Sektor 2 die folgenden Bytes:

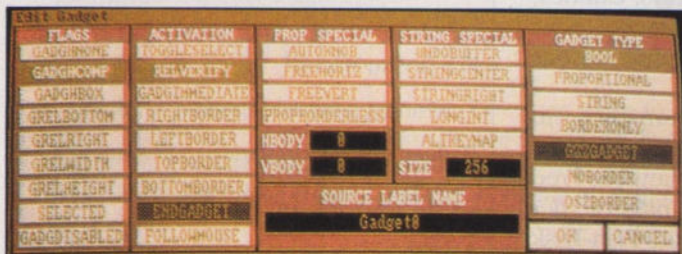


Bild 1. »GadgetEd« kann wahlweise über Menüs oder über Tastatur gesteuert werden

```
$1e8: 8000 in 0000
$1ee: 0280 in 0140
$1fa: 0280 in 0140
```

Die so gepatchte Version des »GadgetEd V2.3« nutzt nun unabhängig von der gewählten Screen-Tiefe immer einen Lores-Screen. Das Programm öffnet, nachdem Sie die Screen-Tiefe im Ausgangsmenü bestimmt haben, den Bildschirm und das Arbeitsfenster mit dem Titel "Work Window".
Dessen Lage und Größe wird wie ein Intuition-Window mit Hilfe der Maus verändert. Erstellen Sie einen Requester, so erscheint zwar ebenfalls ein Fenster, der ausgefüllte Hintergrund signalisiert jedoch den späteren Requester. Die Größe und die Lage können ebenfalls wie bei einem Window modifiziert werden.

Menügesteuert

In der Menüleiste finden Sie rechts die beiden Menüs "FPen" und "BPen". Der unter "FPen" eingestellte Stift wird für alle Zeichenoperationen, wie zum Beispiel: Gadget-Border (Außenrand), oder Text verwendet, während "BPen" die nur selten benötigte Hintergrundfarbe enthält. Der Stift kann einfach durch Anwahl eines anderen aus dem Menü geändert werden.

☐ Das Menü "Project" enthält den obligatorischen About-Menüpunkt, um Programmversion und Autor zu erfahren.

☐ "New" dient zum Löschen aller Gadgets und zum Zurücksetzen der Oberfläche. Sollten Änderungen nicht gespeichert worden sein, führt »GadgetEd« eine Sicherheitsabfrage durch. Es besteht dann keine Möglichkeit mehr, die gelöschten Daten zu restaurieren.

☐ "Load" lädt einen zuvor mit "Save" gespeicherten Entwurf, Voraussetzung ist jedoch, daß es sich um ein gültiges GE-File (siehe oben) handelt.

☐ "Generate Source" erzeugt aus der Oberfläche den Quellcode für C oder Assembler.

☐ Der Menüpunkt "Preferences" öffnet bei Anwahl einen Requester, in dem Sie die Einstellungen des Programms festlegen können.

☐ "Skip zero bit-planes" bedeutet, daß bei Images leere Bitplanes ausgelassen werden. Dies verkleinert die Größe des Image-Puffers, ohne das Aussehen der Grafik zu verändern.

☐ Ist "Auto Gadget -> Image size" aktiviert, so ändert »GadgetEd« die Gadget-Größe beim Laden eines Images in dessen Ausmaße. Sollten Sie jedoch versuchen, dieses Gadget später zu vergrößern, so kann das Programm in der Version 2.3 abstürzen.

☐ Sind "Image Copy" und "Text Copy" aktiviert, so wird nicht nur das Gadget kopiert, sondern auch die zugehörigen Images beziehungsweise Texte.

☐ Möchten Sie, daß die C-Strukturen statisch ("static") sind, so aktivieren Sie das Gadget "Static Structures".
Das "Raw Assembler Source"-Gadget richtet sich an Assembler-Programmierer, die statt Symbolen (zum Beispiel: "SMART_REFRESH") die Werte in hexadezimaler Schreibweise (beispielsweise "$0000") im Quellcode sehen möchten, um keine Include-Files einbinden zu müssen.

☐ Mit Hilfe der beiden Gadgets "WINDOW" und "REQUESTER" können Sie nachträglich ändern, ob der Entwurf ein Fenster oder ein Requester werden soll.

☐ "Save" speichert die Einstellungen in der Datei »devs:gadgeted.prefs«, "USE" übernimmt die Werte. Wählen Sie "Save & Use", so werden die Angaben sowohl in der Datei gespeichert, als auch genutzt.

☐ Der Menüpunkt "Close WorkBench" versucht, die Workbench zu schließen, um so vor allem Chip-Memory zu gewinnen. Dazu dürfen jedoch nur Fenster der Workbench auf dem Workbench-Screen sein; ist beispielsweise ein CLI-Window geöffnet, kann die Workbench nicht geschlossen werden.

☐ Über "Quit", dem letzten Eintrag im Menü "Project", können Sie nach einer Sicherheitsabfrage das Programm verlassen.

Wie funktioniert's?

Um nun ein Gadget zu erzeugen, bewegen Sie den Mauszeiger auf die gewünschte linke obere Ecke des Windows und klicken den linken Mausknopf. Das erscheinende Rechteck legt die Gadget-Ausmaße fest. Haben Sie die rechte, untere Ecke mit dem Mauszeiger erreicht, klicken Sie erneut den linken Knopf, und das Gadget ist definiert. Sollte das Gadget jedoch zu klein sein, so meldet das Programm den Fehler "Gadget to(o) small".
Während dieses Vorgangs gibt »GadgetEd« in der Titelleiste die augenblickliche X- und Y-Position des Mauszeigers, die linke und rechte Ecke des Gadgets sowie dessen Breite und Höhe aus.
Auf die gleiche Art und Weise kann man anschließend weitere Gadgets hinzufügen.
Möchten Sie nun ein Gadget verändern, so sind im Menü "Gadget" die entsprechenden Funktionen zu finden. Deren Handhabung ist einheitlich. Wählen Sie zunächst den gewünschten Menüpunkt an, wodurch das Programm Sie auffordert ("Pick gadget to ..."), das betreffende Gadget anzuklicken. Leider besteht keine Möglichkeit, die Aktion abzubrechen; auch wenn die Aufforderung nach der Anwahl eines anderen Menüpunktes verschwindet, wird sie nach dem Anklicken eines Gadgets ausgeführt.
"Move a gadget" (Tastenkombination: [F1]) dient zum Verschieben eines Gadgets, "Size a gadget" (Tastenkombination: [F2]) erlaubt nachträglich, die Größe zu verändern. Die Funktion "Copy a gadget" (Tastenkombination: [F3]) kopiert das angeklickte Gadget. Je nach »GadgetEd«-Einstellung werden Texte beziehungsweise Images dabei ebenfalls übertragen.
Um ein überflüssiges Gadget aus dem Window oder Requester zu entfernen, wählen Sie "Delete a gadget" (Tastenkombination: [F4]). Ein entferntes Gadget kann nicht mehr zurückgeholt werden. Durch Wählen des Menüpunkts "Edit a gadget" oder durch Drücken von [F5] werden die Flags eines Gadgets geändert.
Diese Angaben umfassen nicht nur die Intuition-Flags, sondern zusätzlich auch »GadgetEd«-interne Werte. Das Feld "FLAGS" umfaßt mehrere Angaben. Die ersten drei Werte bestimmen, was passieren soll, wenn der Anwender das Gadget anklickt. Bei "GADGHNONE" tritt keine Änderung ein, "GADGHCOMP" invertiert das Gadget, und "GADGHBOX" umrahmt das Gadget mit einer Box. "GRELBOTTOM" signalisiert, daß die obere Kante des Gadgets relativ zum unteren Rand des Windows beziehungsweise Requesters ist.
Ein Beispiel: Liegt sie bei gesetzem Flag 100 Pixel über dem unteren Rand und das Window wird vergrößert, bleibt dieser Abstand konstant, und die Y-Position wird verändert.
Ähnlich verhält sich das Flag "GRELRIGHT", nur handelt es sich um den rechten Rand und die X-Ordinate. Die beiden Werte "GRELWIDTH" (Breite) und "GRELHEIGHT" (Höhe) stehen in Zusammenhang mit den Ausmaßen des Windows

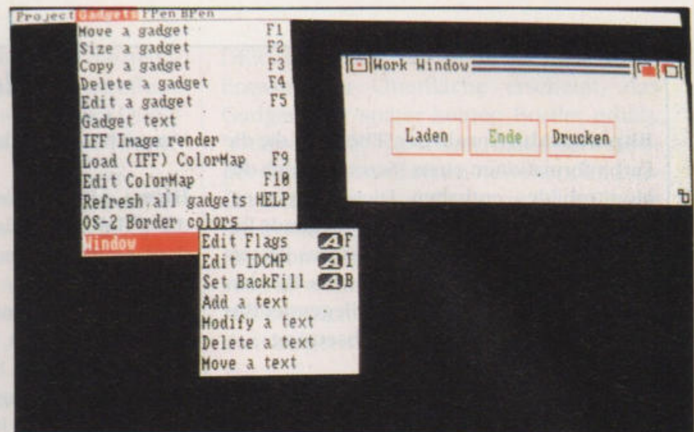


Bild 2. Für jedes Gadget steht eine Vielzahl von Flags zur Verfügung

oder Requesters. Bei "GRELWIDTH" bleibt der Abstand des rechten Gadget-Randes zum rechten Festerrand konstant, "GRELHEIGHT" bezieht sich entsprechend auf den unteren Gadget- und Window-Rand. "SELECTED" bestimmt, daß das Gadget im angewählter Darstellung gezeigt werden soll. Der unterste Wert "GADGDISABLED" desaktiviert das Gadget, das heißt, der Anwender kann es später nicht anwählen. Dieses Flag wird nur im Quellcode gesetzt, da es sonst die Arbeit im Editor mit dem Gadget behindern würde. Unter "ACTIVATION" werden die Flags festgelegt, die später im Quellcode im gleichnamigen Feld der Gadget-Struktur erscheinen.

Flags

"TOGGLESELECT" bedeutet, daß durch Anklicken der Zustand des Gadgets (gewählt/nicht gewählt) sich jeweils ändert und so dem Verhalten eines Kippschalters ähnelt. Bei "RELVERIFY" meldet Intuition dem Fenster, wenn der Mausknopf über einem Gadget betätigt und losgelassen wurde. Bei "GADGIMMEDIATE" unterrichtet Intuition das Window bereits, falls der Mausknopf über einem Gadget gedrückt wurde.
Das oder die betroffenen Flags der vier Border-Felder sollten aktiviert sein, wenn das Gadget sich im Window-Rahmen befindet. "ENDGADGET" bestimmt, daß durch Anwahl des Gadgets der Requester automatisch geschlossen wird. Da dies jedoch nur auf Requester zutrifft, ist diese Option beim Erstellen eines Windows desaktiviert. Das letzte Flag, "FOLLOWMOUSE", bewirkt, daß Intuition Mausbewegungen über dem Gadget nach dem Anklicken meldet. Für einen Schieberegler sind weitere Angaben im Feld "PROPSPECIAL" nötig. Ist "AUTOKNOB" aktiviert, setzt Intuition einen eigenen Mover ein, sonst wird das Image des Gadgets verwendet.

Glossar

Bitplanezahl: Anzahl der "Ebenen", die die Farbinformationen eines "Screens", also des Monitorbildes, enthalten. Diese "Bitplanes" muß man sich wie übereinanderliegende Raster denken, wobei jeder Punkt des Rasters ein gesetztes oder nicht gesetztes Bit darstellt. Eine Reihe übereinanderliegender Bits bilden dann zusammen die Farbsetzung.

Chip-Memory: Speicherbereich, der nicht nur vom Prozessor selbst, sondern auch von den Coprozessoren direkt benutzt werden kann. Dies geschieht über ein kompliziertes System (Multiplex-System), welches entscheidet, wann einer der "Co-Protze" oder der Hauptprozessor auf den Addreßbus zugreifen können. Zugriffe auf Chip-Memory haben jedoch Nachteile: Sie kosten Zeit, da erst entschieden werden muß, welcher Prozessor "gerade darf", und Chip-Memory ist begrenzt, bei vielen älteren Modellen auf 512 KByte, bei den neueren auf 1 MByte, beim A3000 jedoch auf 2 MByte.

Custom-Screen: Bildschirm (also aktuelle Darstellung auf dem Monitor), dessen Parameter vom Programmierer selbst bestimmt werden, während ein Workbench-Screen dem der "normalen" Workbench gleicht.

Dialogbox: Bei bestimmten Programmsituationen ist es erforderlich, daß der Anwender eine Entscheidung trifft, um das Programm in eine bestimmte Richtung zu lenken. Dialogboxen geben hierbei die Möglichkeit, auf einfache Weise (per Mausklick) auszuwählen.

Flags: Hat unser Amiga Probleme, zeigt er Flagge und ruft um Hilfe. So locker kann man eigentlich Flags erklären. Es handelt sich hierbei um bestimmte Bytes einer Speicherstelle, meistens Register eines Co- oder "Nur"-Prozessors (wobei Register interne Speicherstellen dieses Prozessors sind). Ein Flag signalisiert durch Setzen oder Nichtsetzen bestimmte Betriebszustände. Ein Beispiel wäre das Zero-Flag des Akkumulator-Registers (Rechenregister) eines Prozessors. Ist es gesetzt, ist der Inhalt des Akkus gleich null, nicht gesetzt bedeutet dementsprechend, daß der Akku einen Bitwert enthält.

Gadget: Gadget ist ein englischer Begriff, den man eigentlich nicht direkt übersetzen kann. Er bedeutet schlicht und ergreifend "Dingsbums". Gadgets werden zur komfortablen Steuerung von Programmen verwendet, beispielsweise zum Schließen eines Screens, als Schieberegler, zum Blättern im Verzeichnis oder ähnliches mehr.

Hires: Hochauflösender Darstellungsmodus (HIgh RESolution); beträgt auf den Standard-Amigas 640 mal 512 Bildpunkte (Pixel).

IDCMP-Flags: Bestimmen bei Windows über das Aussehen. Einstellbar sind zum Beispiel "kein Rand", Schließ-, Vergrößerungs- und Verkleinerungsmöglichkeit, Verschieben des Bereiches innerhalb des Windows – kurzum alles, was man mit Windows machen kann. Benutzen Sie einmal eines der Workbench-Windows anhand der Funktionen: All dies wird durch die IDCMP-Flags, die eigentlich Schalter für "An" und "Aus" sind, erreicht.

Include-Files: Um die Programmierung des Amiga zu vereinfachen, existieren komplette kleine Programmteile, die von der benutzten Programmiersprache aufgerufen werden können, die Includes. Sie bilden eigentlich den Grundstock für die Systemprogrammierung, weil sie Programmierschritte enthalten, die bei jedem Programm notwendig sind.

Intuition: Die Workbench, mit der wir arbeiten, ist im Grunde auch ein Programm – mit dem Namen »Intuition«

KeyMap: Ein schönes Wort dafür ist "Tastatur-Definitionsdatei". Und im Grunde genommen ist es ganz einfach erklärt: Damit wir in Deutschland unsere Ä, Ö, Ü, ihre kleinen Brüder und das ß auf der Tastatur finden, wird eine "KeyMap" erstellt, die die Zeichen für die aktuelle Taste umdefiniert. Unsere "KeyMap" heißt – natürlich – "d" und befindet sich im logischen Verzeichnis »DEVS:KEYMAPS«.

Label: Ein Label ist eine Sprungmarke; Anwendung findet diese Marke in Programmiersprachen. Ein kleines Beispiel geht aber auch mit CLI-Kommandos:

```
... ask "Programm X laden?"
; if not warn
skip Sprungmarke
endif
; echo "Lade Programm X"
run X
; lab Sprungmarke
echo "hier geht's weiter..."
; ...
```

"warn" ist ein Fehler-Level, der durch "if" abgefagt wird. Ist kein Fehler-Level gegeben ("not warn"), springt das Programm weiter zum Label "Sprungmarke", ansonsten wird erst »X« gestartet.

Lores: Kleinster Darstellungsmodus für Monitore, genannt "LOw RESolution", Bildgröße ist 320 mal 256 Pixel (PAL-Norm).

Pointer: Bestimmte Strukturen, die man beim Amiga zum Programmieren benötigt, brauchen einen festen Platz im Speicher, andere wiederum sind vom System her vorgegeben. Um sie ohne Probleme zu finden, benutzt man "Pointer" (Zeiger), die auf den Beginn dieser Struktur hinweisen. Dies erspart lästiges Suchen oder Neuprogrammieren.

Preferences: Bedeutet sinngemäß "Voreinsteller". Und das sind sie auch; hier kann man den Amiga nach seinen Anforderungen einstellen, zum Beispiel Drucker, Bildschirmfarben, Mauszeiger, serielle und parallele Schnittstelle.

Requester: Ein eigenständiges Window, das vom Benutzer eine Entscheidung erwartet. Auch der "Software-Error" ist einer, man darf dort zwischen Reset und Absturz wählen (etwas scherzhaft gedeutet).

Undo-Puffer: Hat man sich in einer Textverarbeitung entschieden, ein paar Textzeilen zu löschen, und merkt dann, daß auch der Name des Adressaten dabei war, der aus 26 Buchstaben besteht und irgendwie mit "HRWY..." anfing, ruft man für gewöhnlich die Funktion "UNDO" auf. Damit diese arbeiten kann, wird ein bestimmter Teil des Speichers als Puffer benötigt, der die noch nicht richtig gelöschten Daten so lange aufnimmt, bis eine endgültige Entscheidung (Weiterschreiben oder "UNDO") fällt. Einfachstes Beispiel ist auch unser "Trashcan": Erst nach "Empty Trashcan" ist dieser vollständig leer.

USHORT-Array: Feld von ganzen, positiven (unsigned), bytelangen (Short=0–255) Zahlen.

Windows: Eigenständige Teile eines Screens werden als Windows bezeichnet, wenn sie den eigentlichen Screen-Bereich nicht "antasten" und Text- oder Grafikeingaben beziehungsweise -ausgaben zulassen.

Workbench-Screen: Der aktuelle Screen, der nach dem CLI-Befehl "LoadWB" erscheint, also unsere Workbench. Alle Operationen sind bist zum Öffnen eines Custom-Screens auf ihn bezogen

"FREEHORIZ" und "FREEVERT" erlauben das Verschieben des Movers in horizontaler und vertikaler Richtung.
Intuition erzeugt selbständig einen Rahmen um das Gadget, sofern Sie nicht das "PROPBORDERLESS"-Flag setzen. Die Felder "HBody" und "VBody" geben die Anzahl der Schritte in horizontaler und vertikaler Richtung an. Bei einem String-Gadget werden ebenfalls weitere Werte benötigt. Ist "UNDOBUFFER" im Feld "STRING SPECIAL" aktiviert, so erzeugt »GadgetEd« ebenfalls einen Undo-Puffer. "STRINGCENTER" bewirkt, daß der Text in der Mitte des Eingabefeldes dargestellt wird, während bei "STRINGRIGHT" dieser am rechten Rand ausgegeben wird. Ist keine der beiden Optionen aktiviert, so erfolgt die Ausgabe linksbündig. "LONGINT" verwandelt das String-Gadget in ein Integer-Gadget um, in das nur Zahlen, aber keine Buchstaben eingegeben werden können. Falls Sie später eine eigene "KeyMap" für dieses String-Gadget verwenden möchten, muß das Flag "ALTKEYMAP" gesetzt werden. Abschließend enthält "SIZE" die maximale Zeichenanzahl für das betroffene String-Gadget. Der spätere Label im Quellcode wird im Feld "SOURCE LABEL NAME" eingetragen.
»GadgetEd« ersetzt Leerzeichen durch "_"-Symbole, der Name sollte keine Symbolbezeichnung sein, da es sonst beim Compilieren beziehungsweise Assemblieren zu Fehlermeldungen kommen kann. Das letzte Feld, "GADGET TYPE", enthält den Gadget-Typ. Neben den Intuition-Typen "BOOL" (Boolean-Gadget), "PROPORTIONAL" (Schieberegler) und "STRING" (String-Gadget) stehen drei interne »GadgetEd«-Flags zur Auswahl. "BORDERONLY" bewirkt, daß später im Quellcode nur die Border-Struktur erscheint, jedoch keine für das Gadget. Diese muß dann von Programm selbst ins Window gezeichnet werden; bei einem Requester geschieht dies automatisch durch das Betriebssystem. Das Flag kann jedoch nur in Verbindung mit einem Boolean-Gadgets ohne Text und Image gesetzt werden. "NOBORDER" bewirkt, daß der Border nur beim Entwurf der Oberfläche erscheint, das Gadget aber später keinen Border erhält. "OS2BORDER" bestimmt, daß dieses Gadget den aus zwei Farben bestehenden Border erhält, der unter Kickstart 2.0 durch die »gadtools.library« zum Standard wird. Das Intuition-Flag "GZZGADGET" sollten Sie setzen, wenn es sich um ein sogenanntes "GIMMEZEROZERO"-Fenster handelt und das Gadget in dessen Rahmen liegt.

Gadgets "pur" oder "mit"

Sind Sie mit den gewählten Einstellungen einverstanden, so klicken Sie bitte das OK-Gadget. Das Cancel-Gadget dient hingegen zum Abbrechen der Eingabe, die Werte werden in diesem Fall nicht übernommen. Zu jedem Gadget können Texte erstellt werden, indem Sie den Menüpunkt "Gadget text/Add a text" wählen oder [F6] drücken. Haben Sie dann das gewünschte Gadget angeklickt, so erscheint ein Reque-

ster, wobei der gewünschte Text ins obere String-Gadget eingegeben und der sogenannte Draw-Modus (Zeichenmodus) aus der Liste gewählt wird. "JAM1" verwendet nur den im Menü "FPen" eingestellten Stift für die Textdarstellung. Bei "JAM2" wird zusätzlich der Hintergrund mit dem Stift "BPen" überschrieben. Weiterhin kann man die Optionen "COMPLEMENT" und "INVERSVID" setzen.
Bei "INVERSVID" werden beide Stifte vertauscht, während im Complement-Modus die zu setzenden Punkte vorher mit den Pixel der betroffenen Stelle in der Bitplane "exklusiv-oder-verknüpft" werden.

Farbe im Spiel

Wählen Sie "OK", so positionieren Sie mit der Maus den Text relativ zum Gadget, das heißt, der Text kann sich auch außerhalb des Gadgets befinden. Diesen Text können Sie später über "Modify a text" nachträglich ändern, mit "Move a text" verschieben oder durch "Delete a text" entfernen.
Intuition erlaubt alternativ zum Gadget-Border auch das Einbinden von Images. Haben Sie eine Grafik in der IFF-Norm vorliegen (zum Beispiel mit »DPaint« gezeichnet), so können Sie diese direkt mit »GadgetEd« als Image für ein Gadget verwenden. Sie drücken dazu [F7] oder wählen den Menüpunkt "IFF Image render/Load Gadget Image" und klicken nach der Aufforderung das gewünschte Gadget an.
Es erscheint ein Datei-Requester, mit dessen Hilfe Sie den Namen der Grafikdatei auswählen. Sollten Sie bei den programmeigenen Preferences "Auto Gadget -> Image size" eingestellt haben, so ändert »GadgetEd« die Gadget-Ausmaße auf die Image-Größe. Versuchen Sie jedoch, dann die Größe des Gadgets zu ändern, kann es bei der Programmversion 2.3 zum Absturz kommen.
Weiterhin besteht die Möglichkeit, ein zweites Image, das dann beim Anklicken erscheint, durch den Menüpunkt "Load Selected Image" (Tastenkombination: [F8]) ins Gadget einzubauen. Voraussetzung ist aber, daß zuvor ein Gadget-Image bestimmt wurde. Die Funktion "Delete Images" löscht beide Images.
Möchten Sie die Farbpalette einer Grafik in "IFF" übernehmen, wählen Sie den Menüpunkt "Load (IFF) ColorMap" oder drücken [F9]. Der anschließende Menüpunkt, "Edit ColorMap" ([F10]), öffnet einen Color-Requester, in dem Sie die Farben mit Hilfe dreier Schieberegler direkt ändern können. "OK" übernimmt die neue Palette, während hingegen bei "CANCEL" die alte bestehenbleibt. "RESET" setzt alle Farben auf ihre Ausgangswerte zurück, ohne die Dialogbox zu verlassen. Sollen jedoch das Window oder der Requester später auf dem Workbench-Screen erscheinen, so sind die beiden Funktion desaktiviert. "Refresh all gadgets" zeichnet alle Gadgets des Windows neu, alternativ können Sie auch die "HELP"-Taste betätigen.
Ab Kickstart 2.0 werden durch die »gadtools.library« alle Gadget-Border in einer einheitlichen Weise mit einem dreidimensionalen Effekt dargestellt. Dazu werden die obere und linke Begrenzungslinie des Rahmens in einem hellen und die untere und rechte in einem dunklen Ton gezeichnet.
Der Menüpunkt "OS2 Border colors" übernimmt den unter "FPen" gewählten Stift als die helle Farbe und den unter "BPen" gewählten Stift als dunkle Farbe. Wählen Sie nun im Requester "Edit a gadget" das Flag "OS2BORDER", so gibt »GadgetEd« dem Gadget die zweifarbige Umrandung mit den zuvor festgelegten Stiften.
Sollten Sie ein Window erstellen, so heißt der letzte Menüpunkt "Window", ansonsten "Request". Das Untermenü "Window" enthält zusätzlich zwei weitere Punkte. Bei "Edit Flags" können Sie die Flags der New-Window-Struktur ändern.
Diese erscheinen jedoch erst später im Quellcode, da »GadgetEd« sein übliches Fenster benötigt. Der zweite Menüpunkt, "Edit IDCMP", dient zum Festlegen der IDCMP-Flags des Windows. Auch diese Flags werden erst später im Quellcode gesetzt. Die Dialogbox der Funktion enthält aber zusätzlich weitere Felder. "Title" ist der Fenstertitel, und "Label" ist der im Quellcode verwendete Label für die New-Window-Struktur. Ferner können Sie in diesem Requester den "Detail"- und "Block"-Pen des Fensters ändern.

Struktur in Sprache

Den Abschluß bilden die vier Gadgets für die minimalen und maximalen Window-Ausmaße. Alle Werte werden mit "OK" übernommen. Wählen Sie hingegen "CANCEL", so bleiben die alten Einstellungen bestehen. Der Menüpunkt "Set BackFill" übernimmt den "BPen" als sogenannten "BackFill" (Stift, mit dem der Hintergrund gemalt wird).
Dieser ist bei einem Requester direkt in der Struktur integriert, bei Fenstern muß der Hintergrund des Windows mit dem Befehl "RectFill" vom Programm gefüllt werden. Da einige Programmierer (unter anderem auch der Autor von »GadgetEd«) dies gerne in ihre Oberflächen einbauen, hat Jan van den Baard diese Möglichkeit auch beim Entwurf von Windows erlaubt.
Haben Sie nun Ihre Oberfläche entworfen, so wählen Sie im Untermenü "Project/Generate Source" die gewünschte Programmiersprache (C oder Assembler) für den Quellcode aus.
Nach kurzer Zeit erscheint der Datei-Requester zum Eingeben des Dateinamens. Am Ende des C-Quellcodes befinden sich einen Reihe von Konstanten. "NEWWINDOW" oder "REQUESTER" ist der Zeiger auf die New-Window- beziehungsweise Request-Struktur.

Pointer sind wichtig

Sollten Sie Gadgets erstellt haben, so ist "FIRSTGADGET" zusätzlich der Pointer auf das erste Gadget. "FIRSTBORDER" zeigt, sofern man spezielle "BORDERONLY"-Gadgets erstellt hat, auf die erste Border-Struktur. "FIRSTTEXT" ist der Pointer auf den ersten "IntuiText", vorausgesetzt Sie haben Texte an das Window oder den Requester gebunden. Bei Custom-Screen ist außerdem "NEWSCREEN" der Zeiger auf die New-Screen-Struktur, und die Konstante "COLORCOUNT" enthält die Anzahl der Farben im "USHORT"-Array mit der Bezeichnung "Colors".
Die Konstante "WDBACKFILL" enthält den sogenannten Back-Fill-Pen für das Fenster, auch wenn Sie die Funktion "Set BackFill" nicht angewandt haben. Die "USHORT"-Werte der Images werden ebenfalls direkt in den Quellcode eingebunden, müssen jedoch auf jeden Fall im Chip-Memory liegen. Dies können Sie beim SAS/C-Compiler mit Hilfe des Schlüsselwortes "chip" bewerkstelligen.
Als ID (Kennung) für Gadgets verwendet »GadgetEd« den Label des Gadgets, erweitert um "_ID", und definiert ihn im Quellcode vorläufig vor der Gadget-Struktur. Auch der Assembler-Quellcode verfügt über den gleichen Aufbau. Mit Ausnahme von "COLORCOUNT" und "WDBACKFILL" sind die Konstanten des C-Listings nun Zeiger auf Pointer zu der jeweiligen Struktur. ❑

Roger Fischlin (vb)

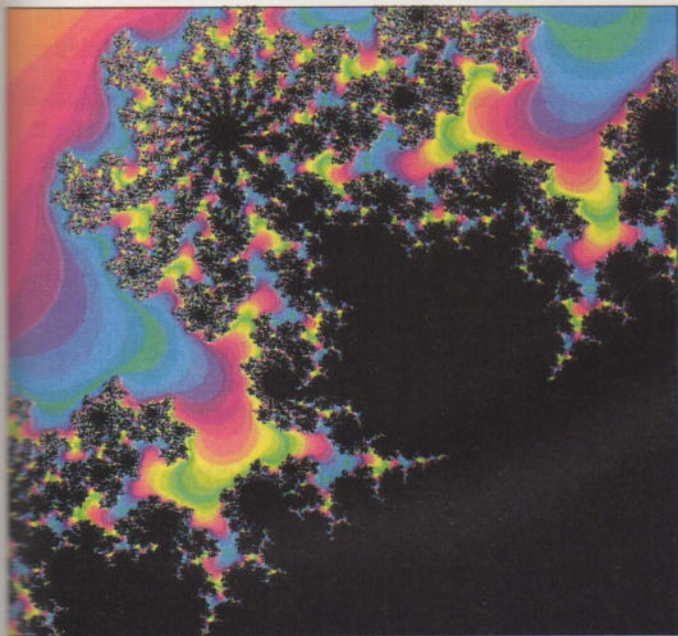
Info:
Name : GadgetEd V2.3
Quelle : Fish #475
Name : Tool-Library
Quelle : Fish #47
Das Programm und die Library finden Sie auch auf der AMIGA-DOS-FD-DISK 3'92.

Grafik

Workshop zu MandelSquare

Wer sich für Apfelmännchen und Julia-Mengen interessiert, ist mit dem Amiga bestens bedient, denn schließlich gibt es hier gute und frei kopierbare Fraktalprogramme beinahe wie Sand am Meer.

Im Moment hebt sich dabei ein Programm besonders hervor, das nicht nur durch seine Schnelligkeit auffällt, sondern auch die Möglichkeit bietet, Animationen zu erstellen: »MandelSquare«. Das Programm stammt vom deutschen Programmierer Olaf Barthel, der den FD-Freaks schon durch seine zahlreichen anderen FD-Programme, wie beispielsweise



»Term«, bekannt sein dürfte (siehe dazu auch AMIGA DOS 2/92, Seite 16 ff.). Wer seine anderen Programme kennt, weiß auch, daß er eine Vorliebe für OS 2.0 hat und seine Programme unter diesem Betriebssystem entwickelt. »MandelSquare« benötigt aber nicht nur OS 2.0, sondern zusätzliche Hardware (Turbokarte, mit 68020, 68030 oder 68040 und einen 68881- oder 68882-Mathe-Coprozessor). Fast-RAM sollte auch nicht fehlen. »MandelSquare« ist so programmiert, daß es die Möglichkeiten des schnellen Hauptprozessors und Coprozessors ausnutzt und dadurch unglaubliche Geschwindigkeiten erreicht. Damit ist das Programm zwar nicht unbedingt für Anfänger und Einsteiger geeignet. Es überzeugt allein schon durch seine Ergebnisse. »MandelSquare« ist auf jeden Fall schneller als andere Mandelbrot-Programme, die nicht direkt auf die Hardware zugreifen. Zudem benutzt »MandelSquare« bei den Berechnungen immer 64-Bit-Zahlen, so daß man sehr genaue Bilder erhält, die trotzdem sehr schnell berechnet werden. Wer einen Amiga 3000 besitzt, braucht sich über die benötigte Hardware keine Sorgen zu machen und kann sofort loslegen.

Während früher die meisten Programme noch jeden Punkt einzeln berechneten, benutzt »MandelSquare« einen leistungsfähigeren Algorithmus, der auch schon von »TurboMandel« (Fish 302) zur Berechnung benutzt wurde. Bei diesem Verfahren werden von der zu berechnenden Bildfläche die vier Eckpunkte berechnet, und dann wird verglichen, ob alle vier dieselbe Farbe haben. Trifft dies zu, so wird die Fläche zwischen den Eckpunkten gefüllt. Sind die Farben unterschiedlich, so wird diese Fläche in vier gleiche Teile geteilt, und die Eckpunkte der kleineren Flächen werden wieder berechnet. Auf diese Weise wird so lange verfahren, bis man eine Fläche gefunden hat, bei der alle vier Eckpunkte gleich sind. Notfalls kann eine Fläche einem Pixel auf dem Bildschirm entsprechen. Der Name des Programms wurde aus dem Englischen abgeleitet (Square = Quadrat).

»MandelSquare« besteht aus dem eigentlichen Programm und einer eventuell vorhandenen Titeldatei. Ist diese Datei, die das Programm unter dem Namen "MandelSquare.Title" im aktuellen Verzeichnis sucht, vorhanden, so berechnet »MandelSquare« beim Programmstart nicht erst das sogenannte Apfelmännchen, sondern verwendet die in dieser Datei gespeicherten Daten. Ist diese Datei nicht vorhanden, so wird zuerst einmal ein Bild berechnet. Da »MandelSquare« dabei auch immer die Dauer einer Berechnung anzeigt, kann man sich von seiner enormen Geschwindigkeit jederzeit überzeugen. Beispielsweise benötigt das zuerst berechnete Bild auf einem Amiga 3000 nur fünf Sekunden. Bei der Berechnung besonders stark vergrößerter Bilder benötigt man mit »MandelSquare« nur einige Minuten; bei anderen Programmen kommen schnell einige Stunden zusammen, wenn diese Programme überhaupt bis zu einer derartigen Rechentiefe vordringen können. Das Programm »MandelBlitz«, eines der wohl schnellsten Programme dieser Art, ist rund zehnmal langsamer und zeichnet ab einer gewissen Tiefe nur noch Linien auf den Bildschirm.

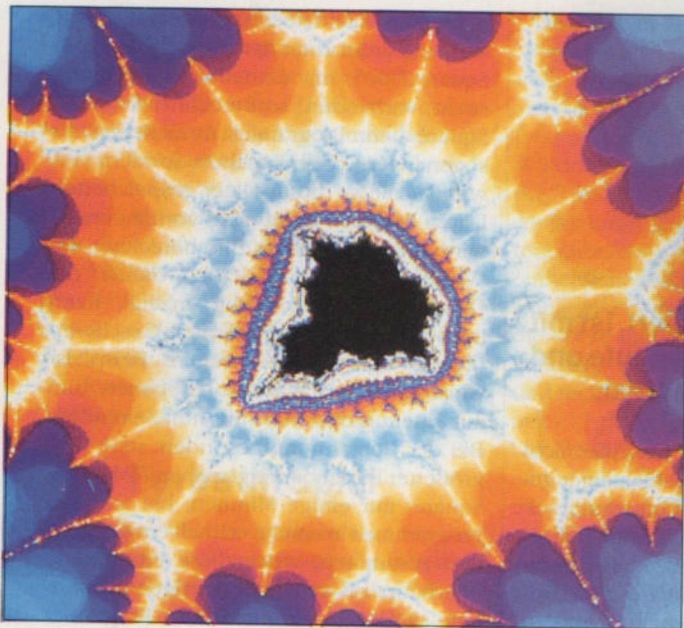
Möchte man mit diesem Programm arbeiten, gelangt man mit der rechten Maustaste in das Pull-down-Menü. Hier können wichtige Parameter verändert und Bilder oder Animationen berechnet werden. Die Iterationstiefe (Rechentiefe) wird im Depth-Menü (zuständig für die Rechentiefe) ausgewählt. Je höher diese Rechentiefe ist, um so genauer werden auch die Bilder. Bei geringen Rechentiefen werden die Bilder zwar schneller berechnet, aber dafür hat man dann auch relativ viele schwarze Flächen auf dem Bildschirm.

Funktionen

Im Programm sind folgende Funktionen enthalten:

– "Open Picture" kann mit »MandelSquare« gespeicherte Bildschirme wieder einlesen. Die oben erwähnte Datei "MandelSquare.Title" kann ebenfalls mit dieser Funktion wieder eingelesen werden.

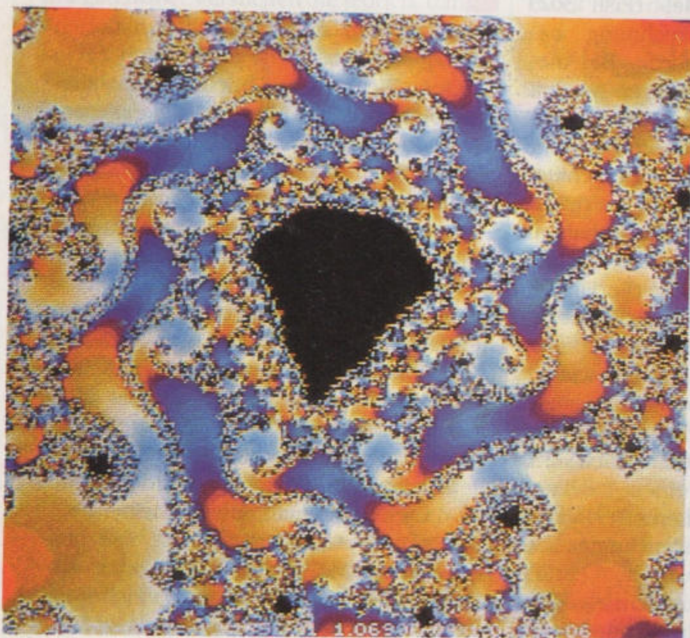
– "Save Picture" speichert den aktuellen Bildschirm unter dem zuletzt eingegebenen Namen ab. Wurde noch kein Name eingegeben, so wird automatisch ein Datei-Requester geöffnet, der auch erscheint, wenn man den Menüpunkt "Save Picture As" anwählt. Mit diesem Menüpunkt kann man auch eine neue "MandelSquare.Title"-Datei anlegen. Bildschirme, die mit »MandelSquare« abgespeichert wurden, sind IFF-Dateien, die man dann mit einem beliebigen IFF-Anzeigeprogramm, wie beispielsweise »Mostra« (Fish 476) oder »PPShow« (Kick 1.3: Fish 542, OS 2.0: Fish 561), anzeigen kann. Auch können diese IFF-



Dateien mit Grafikprogrammen, wie beispielsweise »DPaint«, weiterverarbeitet werden. Wesentliche Daten, die für »MandelSquare« wichtig sind, werden zusätzlich in der IFF-Datei gespeichert. Diese etwas abgewandelten IFF-Dateien werden jedoch von IFF-Programmen als normale IFF-Dateien behandelt. Die Daten, die »MandelSquare« benötigt, gehen bei der Verarbeitung durch andere Programme nämlich verloren.

– "Include Coordinates" fügt bei jedem berechneten Bild an der linken unteren Ecke des Bildschirms die Koordinaten des Bildes ein.

– "Change Resolution" stellt zwei Bildschirmauflösungen zur Verfügung, zwischen denen hin- und hergeschaltet werden kann. Wechselt man die Auflösung, so geht die aktuelle Grafik verloren, und die aktuellen Koordinaten werden in der neuen Auflösung berechnet. Nach dem Programmstart befindet sich »MandelSquare« in der geringen Auflösung, in der die Bilder in 256x256 Pixel berechnet werden (Lores). In der hohen Auflösung, die Hires-Interlace entspricht, werden die Bilder in 512x512 Pixel berechnet. Im Hires-Modus bekommt man zwar genauere Grafiken, aber diese Auflösung hat zwei Nachteile: Im Gegensatz zum Lores-Modus, in dem die Bilder in 32 Farben berechnet werden, können hier nämlich nur 16 Farben verwendet werden. Zudem dauert die Fertigstellung eines Bildes auch etwa fünfmal länger. Für Animationen sollten eigentlich Bilder in der geringen Auflösung ausreichen, da die Animationen dann immer noch ausgezeichnet aussehen. Wer seinen Rechner allerdings auch mal etwas länger arbeiten lassen will, kann auch die hohe Auflösung verwenden. Man sollte dann aber über eine Festplatte und über ausreichend RAM verfügen. Eine kleine Hires-Animation ist nämlich sehr schnell einige MByte lang. Um sich diese Animation dann ansehen zu können, laden Anzeigeprogramme, wie »ShowAnim« oder »PP-Anim«, diese immer in den Speicher, so daß dieser für die Animation auch ausreichend groß sein muß.



– "Magnify" im Calculation-Menü kann auch einzelne Bilder berechnen. Nachdem dieser Menüpunkt angewählt wurde, sieht der Mauszeiger wie ein Fadenkreuz aus. Der Bereich, der vergrößert werden soll, kann dann markiert werden. Dazu bewegt man die Maus in eine Ecke des Bereichs, den man vergrößern will, drückt die linke Maustaste und zieht dann die Maus mit gedrückter linker Maustaste bis zur anderen Ecke des gewünschten Bereichs. Beim Markieren des gewünschten Bereichs können jedoch nur Bereiche ausgewählt werden, die nicht verzerrt sind; dies bedeutet also, daß man nur Quadrate berechnen lassen kann.

Nachdem man die linke Maustaste losgelassen hat, fragt »MandelSquare«, ob der ausgewählte Bereich vergrößert werden soll. Bricht man mit "Cancel" ab, so muß der Menüpunkt "Magnify" erneut angewählt werden, um ein Bild zu vergrößern.

– "Toggle Calculation" bricht die Berechnung einzelner Bilder durch "Magnify" ab. Die Koordinaten des aktuellen Bildschirms sind aber trotzdem korrekt, so daß man auch aus einem nicht kompletten Bild wieder weiter vergrößern kann.

Falls es bei Einzelbildern bei der Berechnung zu Fehlern kommt, kann man aus dem Calculation-Menü die Menüpunkte "Detail" oder "Recalculate" verwenden. Bei beiden Menüpunkten wählt man wie bei "Magnify" einen Bereich mit der Maus aus, der dann neu berechnet wird.

"Detail" berechnet nur die schwarzgebliebenen Flächen und ist auch nützlich, wenn man nach der Berechnung die Rechentiefe verändert hat.

"Recalculate" berechnet Pixel für Pixel und verfügt dabei noch über einen besonderen Modus. Falls der Bereich, der mit "Recalculate" markiert wurde, so groß ist wie der Bildschirm, berechnet »MandelSquare« das Bild im Fast-RAM, da dies schneller geht, als wenn die Berechnungen direkt auf dem Bildschirm sichtbar werden. Jede Zeile, die berechnet wurde, wird aber auf den »MandelSquare«-Bildschirm kopiert.

Animationen

Zur Berechnung einer Animation, wird der Menüpunkt "Begin Animation" ausgewählt. Darauf erscheint ein Datei-Requester, in dem der Name, unter dem »MandelSquare« die Animation abspeichern soll, eingegeben wird. Wie bei "Magnify" wird ein entsprechender Bereich mit der Maus ausgewählt, der in der Animation vergrößert werden soll. Durch einen Requester hat der Anwender dann folgende Möglichkeiten: Klickt man auf "No", so kann nochmals ein Bereich mit der Maus ausgewählt werden. "Abort" bricht die Berechnung der Animation vollständig ab, "Yes" bestätigt, daß dieser Bereich vergrößert werden soll, und das Programm beginnt mit der Berechnung der Animation. Dabei wird das erste Bild, auf dem man den zu vergrößernden Bereich markiert hat, mit in die Animation übernommen. Ist die Animation fertig, kann man mit der Maus den

nächsten Bereich markieren, der vergrößert werden soll. Möchte man keinen weiteren Bereich berechnen, der automatisch an die Animation angehängt wird, so klickt man in dem Requester auf das Abort-Gadget. Eine Animation läßt sich jederzeit mit "End Animation" abbrechen. Die Animationsdatei wird dann geschlossen, wobei das aktuelle Bild aber nicht mit in die Animation übernommen wird.

Wird eine Animation erstellt, versucht »MandelSquare«, diese Animation so zu berechnen, daß die Übergänge zwischen den Bildern nicht zu ruckartig sind. Ab einer gewissen Rechentiefe erscheinen die Übergänge aber dennoch relativ uneben, so daß der Menüpunkt "Smooth Transitions" aktiviert werden sollte. Ist dieser Menüpunkt aktiv, werden einfach mehr Bilder berechnet, so daß die Übergänge wieder weicher werden.

»MandelSquare« ist zwar ein Multitasking-Programm, aber während es rechnet, bemerkt man doch einen deutlichen Geschwindigkeitsverlust für andere Programme. Man sollte also mit einem Task-Monitor, wie beispielsweise »ARTM« (Fish 551), die Priorität von »MandelSquare« auf etwa –30 heruntersetzen. Dadurch wird gewährleistet, daß »MandelSquare« nur dann Rechenzeit in Anspruch nehmen darf, wenn kein anderes Programm Rechenzeit benötigt.

Tips zum Berechnen von Animationen

Möchte man irgendwann eine Animation weiter berechnen lassen, muß man ein wenig tricksen. Dazu speichert man einfach das letzte Bild einer Animation, die gerade beendet wurde, mit dem Menüpunkt "Save Picture As" ab. Soll dann weiter berechnet werden, lädt man dieses Bild wieder ein und beginnt eine neue Animation, so daß zwei Animationen zur Verfügung stehen. Mit dem Programm »CombineANIM«, das sich auf Fish 428 im Verzeichnis "CyroUtils" befindet, können Animationen miteinander verbunden werden. Da »CombineANIM« auch bei schnellen Festplatten relativ lange braucht und die Festplatte dabei sehr häufig benutzt wird, sollte an dieser Stelle die RAM-Disk genutzt werden. Am besten und schnellsten ist es, wenn beide Animationen, die verbunden werden sollen, in die RAM-Disk kopiert werden und man auch »CombineANIM« ins RAM speichern läßt. Hinter Fraktalprogrammen stecken recht ausgeklügelte Rechenalgorithmen, die dafür verantwortlich sind, wie schnell ein Programm die einzelnen notwendigen Werte berechnet. Bei »MandelSquare« hat der Programmierer viel Wert auf Geschwindigkeit gelegt, die durch die entsprechende Hardware natürlich noch unterstützt wird. Bleibt nur noch, viel Spaß im Chaos zu wünschen. ❑

Sandro Paolini (vb)

Info
Name: MandelSquare
Autor: Olaf Barthel
Merkmal: Freeware
Quelle: FRAS-Netz-Mailboxen oder auf der AMIGA-DOS-FD-Disk 3'92

Anwendung

PicBase räumt auf

Bei der Unmenge an schönen Grafiken, die sich monatlich anhäufen, wäre ein wenig Ordnung vonnöten. Ein Freeware-Programm räumt Ihre Bildersammlung auf.

Die Grafikfähigkeiten des Amiga sind hinlänglich bekannt. Auch Amiga-Besitzer, die sich ihren Computer ursprünglich aus "seriöseren" Gründen angeschafft haben, können sich meist nach kurzer Zeit der Faszination der bunten Bilder nicht mehr entziehen.
Eine Konsequenz daraus ist, daß sich unweigerlich früher oder später Unmengen von selbstgemalten oder von FD-Disks stammenden Bildern auf der Festplatte oder in Form von Diskettenstapeln ansammeln.

Ordnung in das Chaos

Da fällt es schwer, die Übersicht zu behalten, und die Suche nach einem bestimmten Bild kann durchaus zu einem kleinen Problem werden. Bis zu einem gewissen Grad sorgen hier Directory-Utilities, wie die kommerziellen Programme »DiskMaster« und »DirectoryOpus« oder in der FD »SID«, für Abhilfe.
Doch selbst bei einer schnellen Festplatte ist es recht zeitaufwendig und unbequem, mit diesen Programmen mehrere Verzeichnisse mit jeweils mehreren Dutzend Bildern zu durchsuchen.

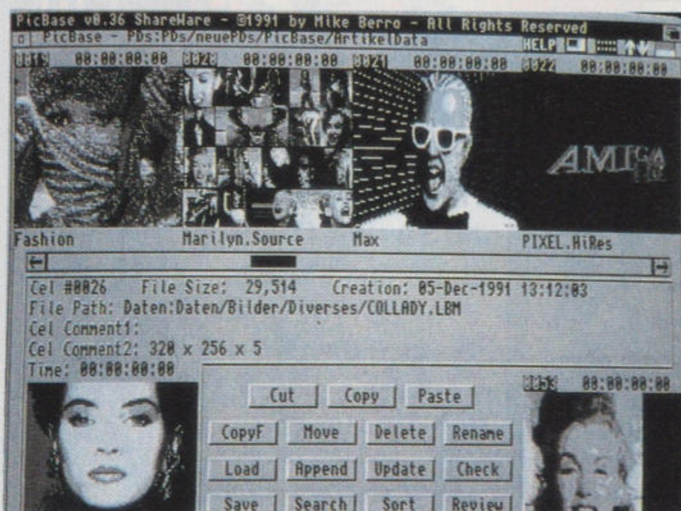


Bild 1. »PicBase« verwaltet Ihre Bilder

Ein Retter in der Not ist – wie so oft – auf einer der Fish-Disks zu finden: Mike Berros Freeware-Programm »PicBase« (Fish #527) ist eine Datenverwaltung speziell für Bilder, die mit einer Besonderheit aufwarten kann: Sie stellt von jedem der ihr bekannten Bilder eine Miniaturausgabe her. Diese Minibilder können dann in großer Zahl gleichzeitig dargestellt werden, was dem Benutzer einen schnellen Überblick über seinen gesamten Bildbestand verschafft.
Um »PicBase« starten zu können, benötigen Sie entweder die Version 2.0 des Betriebssystems oder die »arp-library« (Kickstart-PD 200). Diese Library stellt weiterführende Routinen zur Verfügung, die über die normalen Workbench-1.3-Systemroutinen hinausgehen. Aber noch eine weitere Hürde ist zu nehmen, bevor Sie das Programm tatsächlich benutzen können:
Wie jede Datenverwaltung benötigt »PicBase« natürlich zuallererst Daten – in diesem Falle also Bilder. Die Dateneingabe ist glücklicherweise sehr komfortabel: Nachdem Sie das Gadget "Load" angeklickt haben, können Sie in einem Datei-Requester (Dateianzeiger) den Pfad zu dem Verzeichnis angeben, in dem sich Ihre Bilder befinden. Wenn dabei auch die Bilder in Unterverzeichnissen berücksichtigt werden sollen, müssen Sie das Gadget "All" anklicken, das sich unter dem Datei-Requester befindet. Das Programm liest daraufhin alle Bilder ein und stellt von jedem eine verkleinerte Schwarzweißversion her, die im oberen Teil des »PicBase«-Bildschirms zu sehen ist (siehe Bild 1). Unter dieser Bilderliste befindet sich ein Gadget, mit dessen Hilfe Sie in der Liste vorwärts und rückwärts "blättern" können.
Wenn Sie auf ein Minibild klicken, werden im mittleren Bereich des Bildschirms Informationen über das Originalbild angezeigt: der vollständige Pfad, das Format, die Länge und, falls vorhanden, ein Kommentar. Durch einen Doppelklick auf eines der Minibilder veranlassen Sie das Programm dazu, das Originalbild zu laden und anzuzeigen. Sollte sich das Original auf einer Diskette befinden, die momentan in keinem Laufwerk eingelegt ist, erscheint ein System-Requester, der den Benutzer auffordert, diese Diskette einzulegen. Auf diese Weise können auch Amiga-Besitzer ohne Festplatte von »PicBase« profitieren. Vorausgesetzt, Sie haben Ihre Bilderdisketten konsequent beschriftet, ist es jetzt ein Kinderspiel, ein bestimmtes Bild zu finden.

Alles auf einen Blick

Möglicherweise ist Ihnen aber der Überblick noch nicht groß genug, den Ihnen die Bilderliste, auf der jeweils vier Bilder gleichzeitig zu sehen sind, verschafft. In diesem Fall sollten Sie das Multi-View-Gadget (am rechten oberen Bildschirmrand, links neben den zwei Pfeilen) anklicken.
Dieses aktiviert einen Modus, in dem »PicBase« bis zu 108 (allerdings sehr kleine) Bilder gleichzeitig anzeigen kann (siehe Bild 2). In diesem Modus finden Sie die Informationen über das jeweils angeklickte Bild am unteren Bildschirmrand. Auch hier wird durch einen Doppelklick auf ein Minibild das Original geladen und angezeigt.

Update für die Bilderliste

Über das Gadget "Save" können Sie die Bilderliste abspeichern. Das bedeutet, die Minibilder werden samt Pfadangaben zu den Originalbildern gesichert.
Selbstverständlich läßt sich die Liste aber auch nachträglich erweitern: Über das Gadget "Append" können Sie einzelne Bilder oder auch ganze Verzeichnisse an die Liste anhängen. Benutzen Sie zu diesem Zweck nicht "Load"; damit würden Sie die bereits vorhandenen Daten löschen und eine neue Liste anlegen. "Update" dagegen ist nützlich, um die Bilderliste automatisch auf den neuesten Stand zu bringen.
Diese Funktion überprüft den Inhalt eines im Datei-Requester angegebenen Ver-

zeichnisses und fügt der Bilderliste alle neuen Bilder hinzu. Falls »PicBase« eine neuere Version (erkennbar am Datum) eines Bildes findet, das bereits in der Liste vorhanden ist, wird die ältere durch die neuere Version ersetzt. Das Gadget "Check" schließlich dient dazu, festzustellen, ob Bilder, die in der Liste noch aufgeführt sind, gelöscht wurden. Gegebenenfalls bietet Ihnen »PicBase« dann an, auch die dazugehörigen Minibilder aus der Liste zu entfernen.

Sortieren und Suchen

Über das Gadget "Sort" lassen sich die vorhandenen Bilder nach verschiedenen Kriterien sortieren: nach der Größe, dem Namen, der Pfadangabe (besonders nützlich für Diskettenbenutzer) oder nach dem Datum.
Über die Gadgets "Cut", "Copy" und "Paste" können Sie einzelne Minibilder ausschneiden, kopieren oder an einer anderen Stelle der Liste wieder einfügen. Auf diese Weise können Sie die Reihenfolge der Minibilder in der Liste vollkommen nach Ihren Wünschen gestalten. Ausgeschnittene beziehungsweise zu kopierende Minibilder werden nicht nur in einem Puffer zwischengespeichert, sondern auch auf dem Bildschirm dargestellt (siehe Bild 1, Minibild "Marilyn.Brush").
Diese Operationen beziehen sich nur auf die Minibilder in der »PicBase«-Liste; die Originalbilder sind davon nicht betroffen. Auch sind die vorgenommenen Veränderungen erst dann "endgültig", wenn Sie das Save-Gadget anklicken. Sie können also nach Herzenslust herumexperimentieren, ohne einen Datenverlust befürchten zu müssen.
Eine Suchfunktion ist ebenfalls vorhanden. Sie ist allerdings für ein Datenbankprogramm recht bescheiden ausgefallen: Sie unterstützt keine Joker-Zeichen und verweigert darüber hinaus relativ oft den Dienst. Wenn Sie das Gadget "Review" anklicken, zeigt »PicBase« alle Originalbilder in der Reihenfolge der Liste für jeweils 20 Sekunden an. Mit der linken beziehungsweise rechten Maustaste können Sie ein Bild vor- oder zurückschalten. Da das Programm das jeweils nächste Bild bereits einliest, während das vorige noch angezeigt wird (Double-Buffering), und da der Mauszeiger nicht störend auf dem Bildschirm zu sehen ist, eignet sich diese Funktion hervorragend für Präsentationen.
Das Gadget "Play" spielt die in der Liste vorhandenen Minibilder als Animation ab, was allerdings eher eine Spielerei und nur in den wenigsten Fällen sinnvoll ist.



Bild 2. Im Multi-View-Modus kann »PicBase« bis zu 108 Bilder gleichzeitig anzeigen

Im Hintergrund

»PicBase« ist Ihnen jedoch nicht nur beim Katalogisieren und Wiederfinden von Bildern behilflich: Sie können mit Hilfe dieses Programms Ihre Bildersammlung auch reorganisieren, sprich Bilder löschen, kopieren, verschieben oder umbenennen. Komfortablerweise beziehen sich diese Funktionen nicht nur auf einzelne Bilder, sondern wahlweise auch auf Gruppen von bis zu 20 Bildern, die zu diesem Zweck im Clipboard (am unteren Bildschirmrand) abgelegt werden können.
»PicBase« kann auch mit einem Iconify-Modus aufwarten.
Ein Klick auf das entsprechende Gadget bewirkt, daß der eigentliche Programmbildschirm geschlossen wird. Das spart etwa 100 KByte kostbares Chip-Mem. Dafür öffnet sich auf der Workbench ein neues Fenster, das Sie anklicken können, um »PicBase« wieder zu aktivieren.
Die Help-Funktion des Programms sollten sich einige kommerziell vertriebene Programme zum Vorbild nehmen: Zu jedem auf dem Bildschirm vorhandenen Gadget kann der Benutzer einen erläuternden Text abrufen. Dies erspart so manchen Griff zur englischsprachigen Anleitung.

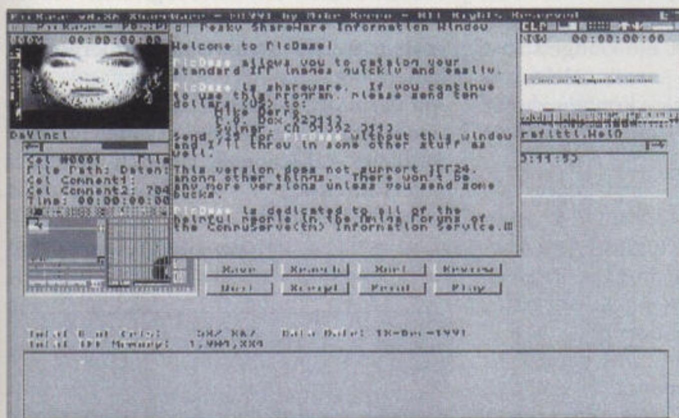


Bild 3. Für das Zusammenstellen von Bilder-Disketten ist »PicBase« unentbehrlich

»PicBase« sorgt für Durchblick

Ob Sie nun eine Festplatte besitzen oder nicht – ab einer gewissen Anzahl von Bildern kommen Sie gar nicht darum herum, auf irgendeine Art Ordnung in Ihre Bildermassen zu bringen. Lassen Sie doch »PicBase« diese Aufgabe für Sie erledigen. In der vorliegenden Version unterstützt das Freeware-Programm "lediglich" IFF-Bilder.
Läßt man dem Programmierer Geld zukommen (liegt im eigenen Ermessen der Benutzer), können weitere Versionen erscheinen, die auch andere Bildformate (IFF24, DCTV, HAM-E) verarbeiten können. ❑

Hartmut Schumacher (vb)

Info
Name: PicBase
Merkmal: Freeware
Autor: Mike Berro
Quelle: Fish-Disk 527

Tool

DeskBench – Komfort für Tintenspritzer

Seit Tintenstrahldrucker wie der HP Deskjet in akzeptable Preisbereiche gerutscht sind, nehmen sie immer mehr den Platz der Nadeldrucker ein. Das Programm »DeskBench« macht Deskjet-Besitzern das Leben jetzt noch leichter.

Die Deskjet-Tintenstrahldrucker von Hewlett Packard weisen ein hervorragendes Preis-Leistungsverhältnis auf. Sie eignen sich somit ideal für Amiga-Besitzer, die Wert auf sehr gute Ausdrucke legen. Ein Schwachpunkt dieser Drucker ist jedoch die Tatsache, daß ihre internen und ihre zusätzliche Schriften nicht über das druckereigene Bedienfeld ausgewählt werden können. Selbst preisgünstige 9-Nadel-Drucker sind in dieser Hinsicht benutzerfreundlicher. Deskjet-Besitzer sind also darauf angewiesen, daß ihre Software die Ansteuerung der Schriften vornehmen kann – doch leider sind viele Programme dazu nicht in der Lage.

Das Freeware-Programm »DeskBench« macht diesem mißlichen Zustand nun ein Ende: Es ermöglicht es nicht nur, die zu verwendende Schriftart anzuwählen, sondern auch auf eine Vielzahl von weiteren Druckereinstellungen Einfluß zu nehmen, und zwar bequem über eine mausgesteuerte Benutzeroberfläche (daher auch der Name des Programms: eine Workbench für den Drucker Deskjet = »DeskBench«).

Auf dem »DeskBench«-Bildschirm (siehe Bild 1) befindet sich eine Reihe von Gadgets, die Sie anklicken können, um die verschiedenen Einstellungen zu verändern, als da sind: die Schriftart, die Schriftgröße, die Anzahl der Zeichen pro Zoll, die Schriftlage, die Druckausrichtung, die Druckqualität, die Stärke der Zeichen, die Anzahl der Zeilen pro Zoll, der Zeichensatz und die zu verwendende Druckerschnittstelle. Darüber hinaus können Sie auch einen Seitenvorschub auslösen und den Selbsttest des Druckers aufrufen.

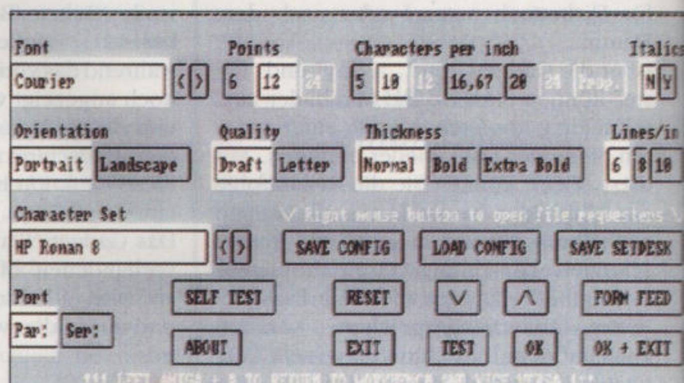


Bild 1. »DeskBench«: Die totale Kontrolle über Deskjet-Drucker

Einfachste Bedienung

Nicht jede Druckereinstellung läßt sich beim Deskjet-Drucker jedoch beliebig mit jeder anderen kombinieren (beispielsweise ist bei querformatiger Druckausrichtung keine Kursivschrift möglich). »DeskBench« erkennt dies und markiert die Gadgets, die Sie jeweils nicht anwählen können, durch eine weiße Schriftfarbe.

Über das Gadget "SAVE CONFIG" können Sie sämtliche Druckereinstellungen in einer Konfigurationsdatei abspeichern. Diese Datei wird beim Programmstart geladen und stellt automatisch die bevorzugte Druckerkonfiguration ohne lästiges Herumklicken zur Verfügung. Am dankbarsten werden Deskjet-Besitzer jedoch für die Möglichkeit sein, sogenannte "SetDesk-Dateien" abzuspeichern. Diese Dateien enthalten die mit »DeskBench« festgelegten Druckereinstellungen und können mit dem Befehl »Execute« in der Shell (oder schon in der »startup-sequence«) aufgerufen werden.

Multitasking?

Auf diese Weise macht sich »DeskBench« sozusagen selbst überflüssig: Erstellen Sie einfach für jede der Druckerkonfigurationen, die Sie häufig benötigen, eine "SetDesk-Datei", die Sie dann bei Bedarf aufrufen können, um den Drucker zu konfigurieren, ohne erst das (nicht gerade kurze) Hauptprogramm laden zu müssen. »DeskBench« ist ein kompiliertes und komprimiertes AMOS-Programm. Wie jedes in dieser Sprache geschriebene Programm ist »DeskBench« zwar Multitasking-fähig, es verlangsamt jedoch andere gleichzeitig verwendete Programme merkbar. Es ist daher nicht sehr sinnvoll, »DeskBench« ständig im Hintergrund laufen zu lassen. Statt dessen sollten Sie das Programm wieder beenden, nachdem Sie den Drucker konfiguriert oder "SetDesk-Dateien" hergestellt haben.

Auf der AMIGA-DOS-FD-Diskette 3'92 befindet sich neben dem Programm eine ausführliche Anleitung in Deutsch, Englisch und Französisch. ❑

Hartmut Schumacher (vb)

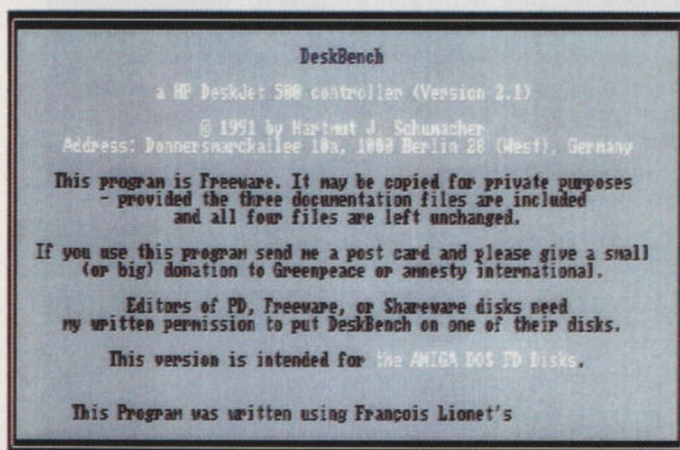


Bild 2. »DeskBench« – Das Drucker-Utility

Info
Name: DeskBench
Autor: Hartmut Schumacher
Merkmal: Freeware
Quelle: AMIGA-DOS-FD-Disk 3'92

FD-Tips

Das Hilfsprogramm MachIII

Utilities erleichtern das Anwenderleben ungemein. Wir liefern Ihnen Tips und Tricks zum beliebten Vielzweck-Utility.

Eines der verbreitetsten Hilfsprogramme ist ohne Zweifel »MachIII« von Brian Moats (Fish-Disk 471). Es vereinigt in sich eine Fülle von Funktionen, die sich beim täglichen Umgang mit der Workbench als äußerst nützlich erweisen, unter anderem:

- einen Mausbeschleuniger;
- automatische Fensteraktivierung;
- bequemes "Durchblättern" von Fenstern und Bildschirmen;
- einen Bildschirmschoner;
- eine Uhrzeit- und Datumsanzeige samt Wecker;
- eine programmierbare Funktionstastenbelegung;
- Tastatur- und Mausmakros.

Während die Maus- und Uhrfunktionen wohl von jedem »MachIII«-Anwender genutzt werden, schreckt die auf den ersten Blick kompliziert erscheinende Makrofunktion viele Benutzer ab.

Dabei sind Makros ganz einfach Folgen von Tastatureingaben oder Mausbewegungen, die gespeichert werden und die der Benutzer dann bequem mit lediglich einem Tastendruck wieder aufrufen kann. Ein spielerisches Verändern der Beispielkonfigurationsdatei macht schnell mit dem grundlegenden Prinzip vertraut und befähigt dazu, Makros zu schreiben, die den eigenen Bedürfnissen entsprechen.

Aber »MachIII« kann noch viel mehr: Es ist möglich, Makros zu einem bestimmten Zeitpunkt oder in bestimmten Zeitabständen automatisch starten zu lassen. Dies eignet sich hervorragend dazu, Programme mit einer Auto-Save-Funktion zu versehen. Sie kennen eine solche Funktion wahrscheinlich von Programmen wie »WordPerfect«: In vom Benutzer festlegbaren Zeitabständen speichert das Programm selbständig eine Sicherheitskopie der eingegebenen Daten (in diesem Beispiel des Textes) ab. Wenn Ihr Amiga dann abstürzen sollte, ist nur ein relativ geringer Teil Ihrer Arbeit verloren, da Sie ja auf die Sicherheitskopie zurückgreifen können.

Was machen Sie nun aber, wenn Sie diese Auto-Save-Funktion schätzen gelernt haben, aber einige weniger fortschrittliche Programme nicht mit ihr aufwarten können? Sie haben es erraten: Auch hier erweisen sich die »MachIII«-Makros als hilfreich. Wenn Sie ein Makro mit der Zeichenfolge

```
\a+x
```

einleiten, wird dieses Makro alle x Minuten ausgeführt. Das Makro

```
\a+5\RS
```

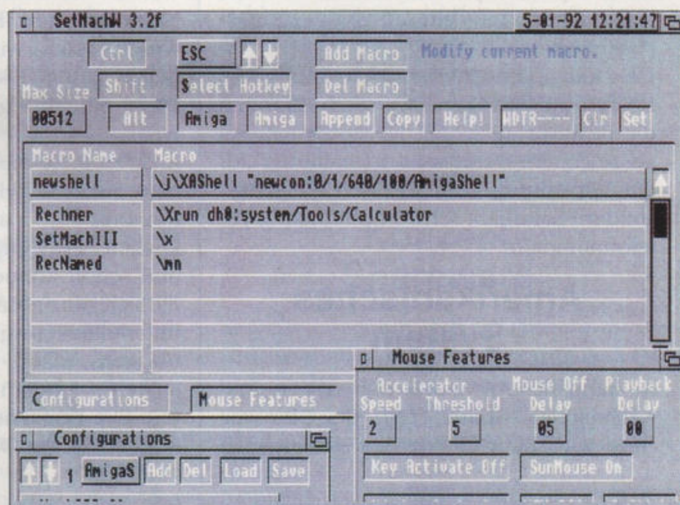
würde also automatisch alle fünf Minuten die Tastenkombination [rechte Amiga-Taste] und [S] betätigen.

Diese Tastenkombination bewirkt beispielsweise bei dem Programm »IconEdit« (aus dem Verzeichnis "Tools:" der Workbench 2.0) ein Abspeichern des gerade bearbeiteten Icons. Bei anderen Programmen werden Sie andere Tastenkombination verwenden müssen. In welcher Form die diversen Qualifier-Tasten (rechte und linke Amiga-Taste, Control, Shift, Alt) oder auch Mausbewegungen in Makros eingetragen werden, können Sie in der Anleitung nachlesen oder aber sich auflisten lassen, indem Sie das Gadget "Help!" des Konfigurationsprogramms »SetMachIII« anklicken.

Die Zahl 5 im obigen Beispiel sollten Sie durch die Anzahl der Minuten ersetzen, die zwischen zwei Aufrufen der Abspeicherfunktion liegen soll. Welche Zahl hier sinnvoll ist, hängt unter anderem von Ihrem Arbeitstempo ab.

Solche automatischen Makros sollen natürlich nur dann tatsächlich ausgeführt werden, wenn Sie gerade mit dem entsprechenden Programm arbeiten. Nutzen Sie daher die Möglichkeit, mehrere verschiedene »MachIII«-Konfigurationen zu verwenden, die aktiviert werden, wenn sich der Mauszeiger auf einem bestimmten (namentlich anzugebenden) Fenster oder Bildschirm befindet (siehe Kapitel 7 der Anleitung). Diese Vorgehensweise empfiehlt sich auch, wenn Sie Programme wie »WordPerfect« benutzen, die sich beispielsweise mit der Funktionstastenbelegung oder der Funktion "SunMouse" nicht vertragen. Bei der Definition mehrerer Konfigurationen ist das Programm »AllTitles« sehr nützlich, das im Lieferumfang der registrierten Version von »MachIII« enthalten ist und die Titel aller Bildschirm und Fenster auflistet.

Und hier zu guter Letzt ein grundlegender, aber sehr wirkungsvoller Tip: Bestellen Sie die registrierte Version von »MachIII«. Sie besitzt einige neue Eigenschaften und weist im Gegensatz zu den unregistrierten Versionen kein störendes Anfangsfenster auf. Registrierte Benutzer in Deutschland, Österreich und der Schweiz erhalten darüber hinaus zusammen mit ihrer registrierten Version Hans Röcknagels deutsche Übersetzung der Originalanleitung. Die neueste frei vertreibbare (also unregistrierte) Version 3.2 von »MachIII« finden Sie auf der zu diesem Heft gehörenden FD-Diskette. ❑



Das vielseitige Utility »MachIII« in seiner neuesten Version

Hartmut Schumacher (vb)

Info
Name: MachIII
Version: 3.2
Shareware-Gebühr: 15 US-$ (ca. 30 DM)
Autor: Brian Moats
Quelle: AMIGA-DOS-FD-Disk 3'92

Anwendung

Telefonieren mit Addresser

Verwaltungsprogramme sind eine sehr nützliche Sache, wenn es um große Datenmengen geht.

Geht es Ihnen auch so, daß Sie sogar die Telefonnummer Ihres besten Freundes erst aus einem Notizbuch hervorkramen müssen, weil Sie einfach ein gestörtes Verhältnis zu Zahlen im allgemeinen und zu Telefonnummern im besonderen haben?
Dann werden Sie sicher das Programm »Addresser« genauso zu schätzen wissen wie ich: Gefunden auf Fred Fish 559, liegt das Programm in der Version 1.0 vor. Der Autor ist Jeff Kelly. »Addresser« ist – wie der Name schon sagt – ein Adreßverwaltungsprogramm.
Dies klingt ja noch nicht allzu aufregend; interessant wird es erst, wenn ich jetzt noch die Features aufzähle, die das Programm auszeichnen: »Addresser« ist nicht nur auf die Verwaltung von Adressen ausgelegt, sondern handhabt schwerpunktmäßig Telefonnummern.

Amerikanisches System

So kann der Anwender neben Vor- und Nachnamen, Straße, Ort, Staat/Provinz (hier wird deutlich, daß es sich um ein amerikanisches Programm handelt), Land und ZIP (entspricht in etwa unserer Postleitzahl) auch noch zwei Telefonnummern angeben: die private und die geschäftliche Rufnummer der eingetragenen Person (siehe Bild 1).
»Addresser« bietet Ihnen nun die Möglichkeit, wahlweise eine der beiden Nummern automatisch anzuwählen.
Hierzu muß lediglich ein Modem an die serielle Schnittstelle des Amiga angeschlossen und mit der Telefonleitung verbunden sein. Nachdem Sie sich entschieden haben, welche der beiden Nummern gewählt werden soll, wird diese automatisch vom Modem gewählt.
Anschließend nehmen Sie den Telefonhörer ab, klicken auf "OK", und schon können Sie sich ganz normal mit Ihrem Gegenüber unterhalten (vergleiche Bild 2). Sollten Sie übrigens schon Tonwahl besitzen oder sollte Ihr Modem nicht Hayes-kompatibel (Modem-Kommando-Standard) sein, so können Sie den Wählbefehl angeben; voreingestellt ist dabei "Atdt". Ferner können Sie zu jedem Eintrag eine kleine Bemerkung hinzufügen.
Das ist an sich nichts Besonderes, folgendes Feature hingegen schon: »Addresser« besitzt einen Schalter, der mit "Mark" beschriftet ist. Dieser kann individuell für jeden einzelnen Datensatz gesetzt werden, um diesen zu markieren. Diese Option ist ideal geeignet, um die wichtigsten Adressen zu markieren. Es gibt nämlich einen besonderen Anzeigemodus, der Sie nur mit den so gekennzeichneten Datensätzen konfrontiert.
Selbstverständlich kann man mit »Addresser« Datensätze ändern, hinzufügen und auch löschen. Das sollte bei dieser Art von Programmen eigentlich zum Standard gehören. Gleiches trifft auch für die Search-Funktion (Suchfunktion) zu, doch auch hier hebt sich »Addresser« vom Durchschnitt ab.

»Addresser« wählt automatisch

Der Anwender kann jedes beliebige Feld (Name, Straße, Bemerkung und ähnliches) nach einem bestimmten String (Zeichenkette) durchsuchen. Interessant ist auch hier wieder die Option "Search & Mark", die jeden passenden Datensatz durchsucht und markiert (Bild 3).
Auch die Druckfunktion wird von »Addresser« hervorragend gelöst: Man kann sich die Daten entweder tabellarisch, als Telefonliste, als Adreßaufkleber oder als Adreßkarten ausdrucken lassen. Gerade bei den letzten beiden Arten wirkt es sich für den deutschen Anwender wiederum negativ aus, daß das Programm auf die amerikanischen Bedürfnisse abgestimmt ist: Die so gewonnenen Aufkleber und Karten sind für den eigentlichen Einsatz praktisch unbrauchbar. Für den Ausdruck von Daten in tabellarischer Form oder als Telefonliste ist die Druckfunktion aber uneingeschränkt nutzbar.

Hallo Drucker

Damit wären wir schon bei einer weiteren Besonderheit des Programms, dem "Mailmerge". »Addresser« bietet die Möglichkeit, Adressen als ASCII-Datei, als Wordperfect-

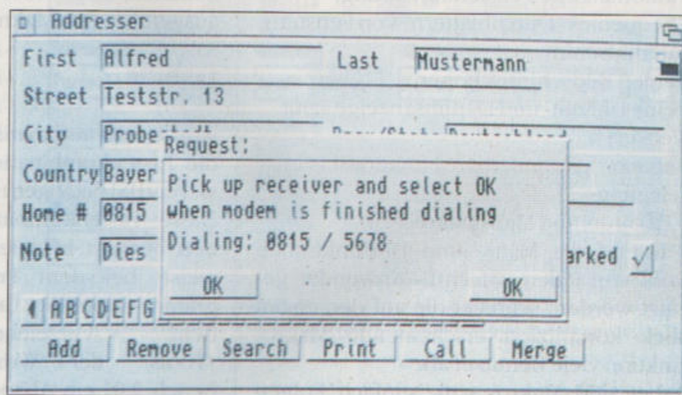


Bild 2. »Addresser« bei der Telefonwahl über Modem

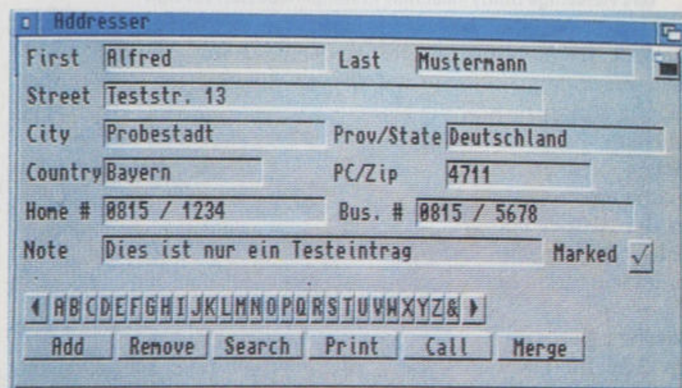


Bild 1. Der Hauptbildschirm von »Addresser«

Datei oder als Prowrite-Datei abzuspeichern. Sie können den Dateinamen dabei direkt eingeben oder aber auch einfach über den ARP-Requester (eine Dateiauswahl, die durch die Nutzung der »arp.library« über wesentlich mehr Funktionen verfügt; siehe dazu auch Computer-Lexikon auf Seite 112) auswählen.
Selbstverständlich können Sie hier wie bei der Druckfunktion wählen, ob Sie alle Adressen, nur die markierten Sätze oder nur den aktuellen Datensatz abspeichern wollen (Bild 4). Mit dieser Funktion können Sie die eingegebenen Adressen also direkt in Ihre Briefe oder ähnliches übernehmen. Allerdings muß auch hier wieder umgestellt werden, um die gewohnten deutschen Anschriften zu erhalten.

Einfache Bedienung

Nach diesem Streifzug durch die Funktionen des Programms möchte ich noch auf die Bedienung eingehen. Im Fenster von »Addresser« finden sich einige Gadgets, mit denen innerhalb der Datensätze geblättert werden kann.
So können Sie mit den Proportional-Gadgets vor- und zurückblättern. Ferner finden sich weitere Gadgets, die mit den 26 Buchstaben des Alphabets beschriftet sind. Damit können Sie direkt zum ersten Datensatz springen, dessen Nachname mit dem jeweiligen Buchstaben beginnt – eine Funktion, die von den Tischkarteien abgeschaut wurde und wirklich hilfreich ist!
Daß das Programm gründlich durchdacht ist, zeigt sich allein an der Tatsache, daß ein Gadget ("&") vorgesehen wurde, mit dem alle Datensätze angesprungen werden, deren Namen nicht mit einem Buchstaben beginnen.
Ferner findet sich rechts oben ein Schalter, mit dem das Programm "iconifiziert" werden kann: Das Fenster wird geschlossen, es bleibt lediglich eine kleine Leiste auf der Workbench übrig.
Bei Bedarf klickt man diese an, und »Addresser« ist wieder aktiv. Dieses Verfahren spart Speicher und dient der Übersichtlichkeit. Das Programm läßt sich auch gleich "iconifiziert" starten, was gerade für einen Aufruf aus der »startup-sequence« heraus ideal ist.
Nachdem »Addresser« beendet wurde, wird die Adreßdatei auf dem Datenträger aktualisiert und bei jedem weiteren Start automatisch eingelesen.
Hier wird sich manch einer vielleicht eine flexiblere Dateiverwaltung wünschen, aber ich stehe auf dem Standpunkt, daß man gerade bei einer Adreßverwaltung nur eine Datei benötigt. Alle angesprochenen Funktionen lassen sich außer über Gadgets auch per Menü steuern, wobei für jeden Menüpunkt eine Tastenkombination existiert, was wirklich äußerst benutzerfreundlich ist.

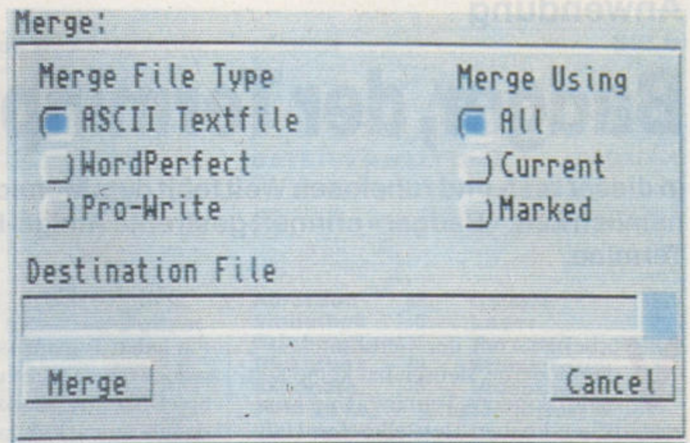


Bild 4. Merge-Funktion von »Addresser«

Immer aktuell

Obwohl unter Kickstart 1.3 entwickelt, läuft »Addresser« auch anstandslos mit Kickstart 2.04 zusammen. Lediglich wenn ein anderer Font eingestellt wurde, können die Gadgets etwas "aus dem Rahmen fallen".
»Addresser« ist ein hervorragendes Programm, das ich uneingeschränkt jedem Anwender empfehlen kann. Negativ wirkt sich lediglich die Tatsache aus, daß das Programm auf amerikanische Verhältnisse zugeschnitten wurde, was sich bei ein paar Funktionen (Druck von Adreßaufklebern, Mailmerge) hier in Deutschland nicht gut macht.
Da aber wirklich nur wenige Funktionen davon betroffen sind, fällt diese Tatsache meines Erachtens nicht so sehr ins Gewicht. Sicher kann man mit jeder Dateiverwaltung (wie beispielsweise dem kommerziellen Programm »SuperBase«) eine eigene Adreßverwaltung programmieren, aber dies ist relativ aufwendig.
»Addresser« bietet wirklich einige Funktionen, die dieses Programm deutlich über den Durchschnitt heben; die (englische) Anleitung weist den Anwender anschaulich in das Arbeiten mit dem Programm ein. Interessant ist auch das Shareware-Konzept des Autors Jeff Kelly:

Konzept Shareware

Er weist darauf hin, daß das Programm frei vertrieben werden darf, erwartet jedoch von den Anwendern eine Spende von 10 US-Dollar. Dafür erhält der Anwender dann auch entsprechend immer die neueste Version.
Es bleibt nur zu hoffen, daß dieses Programm noch weiterentwickelt wird, da es bereits in der Version 1.0 erstaunlich professionelle Züge aufweist. ❑

Frank Gärtner (vb)

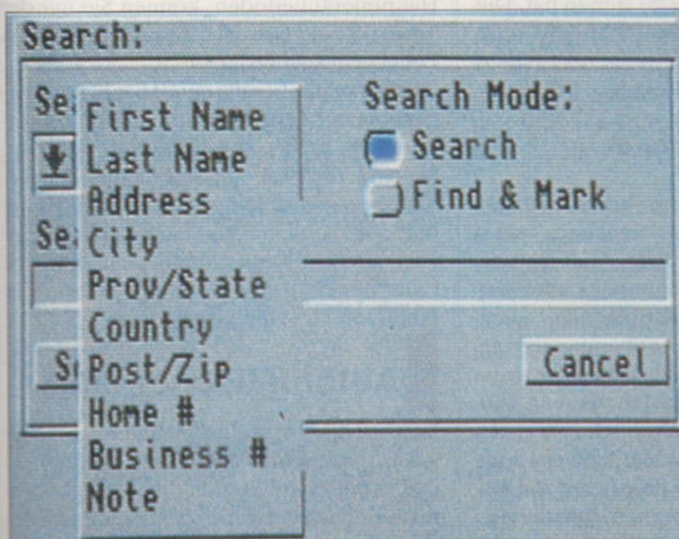


Bild 3. Suchmodus mit Auswahl des Suchfeldes

Info
Name: Addresser
Autor: Jeff Kelly
Merkmal: Shareware, Gebühr 10 US-$
Quelle: Fish-Disk 559

Anwendung

Badger, der Terminplaner

In dieser rast- und ruhelosen Welt fehlt dem Menschen etwas, nämlich Zeit. »Badger« erinnert gestreßte Amiga-Besitzer an ihre Termine.

Vorbei ist's mit der Geruhsamkeit – der moderne Mensch hat keine Zeit mehr, sondern Termine. Und dementsprechend haben Hersteller von Hilfsmitteln wie Terminkalendern Hochkonjunktur. Nun kann man sich zwar darüber streiten, ob Computer ihren Benutzern tatsächlich Zeit sparen, aber es müßte doch zumindest möglich sein, mit dem Amiga und einem entsprechenden Programm Ordnung in das Terminchaos zu bringen?
Dies gelingt auch; beispielsweise mit George Kerbers Shareware-Programm »Badger«, das Sie zuverlässig an Termine erinnert und dabei recht einfach zu bedienen ist.
Sie können in diesen elektronischen Kalender eine Reihe von Terminen eintragen, zusammen mit einem erläuternden Text und einer Angabe, wie viele Tage vor dem eigentlichen Termin Sie an ihn erinnert werden möchten.
Beim Booten Ihres Amiga kann »Badger« dann automatisch aufgerufen werden und informiert Sie über alle Termine, deren "Vorwarnzeit" bereits begonnen hat.
»Badger« arbeitet dabei datumsorientiert, kann also keine Termine verwalten, die an eine bestimmte Uhrzeit gebunden sind. Auch sollten Sie bedenken, daß es nur dann sinnvoll ist, »Badger« einzusetzen, wenn Sie Ihren Amiga relativ häufig benutzen. Dank der Vorwarnzeit, die bis zu 45 Tage betragen kann, ist es zwar nicht nötig, ihn jeden Tag anzuschalten, wenn Sie ihn jedoch nur alle zwei Wochen mit einem Spiel füttern, ist »Badger« eindeutig nicht das geeignete Programm für Sie.

Maus ade

Sie können »Badger« über die Workbench durch einen Doppelklick auf sein Icon oder aus der Shell heraus aufrufen. Daraufhin öffnet sich ein Fenster auf der Workbench, in dem das Hauptmenü des Programms zu sehen ist (siehe Bild 1). Hier, wie in allen weiteren »Badger«-Menüs, können Sie die gewünschten Programmfunktionen anwählen, indem Sie über die Tastatur entweder die zugeordnete Ziffer oder aber den Buchstaben eingeben, der in der Funktionsbeschreibung farbig markiert ist. Eine Unterstützung der Maus ist nicht vorgesehen.
Wenn Sie das Programm zum erstenmal starten, werden Sie höchstwahrscheinlich zunächst einmal die wichtigsten Termine eintragen wollen. Zu diesem Zweck können Sie im Hauptmenü auf der Tastatur den Buchstaben "a" oder die Zahl "2" betätigen. Daraufhin werden Sie vom Programm aufgefordert, das Datum des Termins einzugeben. Hierbei müssen Sie beachten, daß – amerikanischen Konventionen folgend – der Monat vor dem Tag zu stehen hat. Die Eingabe für den 22. November 1991 muß also folgendermaßen aussehen:

```
11/22/91
```

Wenn Sie keine Jahreszahl angeben, setzt »Badger« automatisch das aktuelle oder das darauffolgende Jahr ein (abhängig davon, ob das Datum im aktuellen Jahr bereits verstrichen ist).
Anschließend erwartet das Programm die Information, wieviel Tage vor dem eigentlichen Termin Sie an ihn erinnert werden möchten. Eine solche Vorwarnzeit kann bis zu 45 Tage lang sein.
Es ist nur dann sinnvoll, die nächste Frage – nach einem Referenzjahr – zu beantworten, wenn es sich bei dem Termin um einen Geburts- oder um einen Jahrestag handelt. Bei allen anderen Arten von Terminen können Sie diese Frage mit der Return-Taste überspringen.
Im Anschluß daran erkundigt sich das Programm nach der Beschreibung des Termins. Der Text, den Sie hier eingeben, wird ausgegeben, wenn »Badger« Sie an den Termin erinnert. Soll das Programm bei Geburts- und Jahrestagen anzeigen, wie alt die entsprechende Person wird beziehungsweise der wievielte Jahrestag stattfindet, muß in dieser Beschreibung das Wort "Birthday" oder "Anniversary" auftauchen. Auch wenn Sie Wert darauf legen, daß »Badger« in Ihrer Muttersprache mit Ihnen spricht und die Beschreibung des Termins daher auf deutsch eingegeben wird, kommen Sie nicht umhin, die englische Bezeichnung für die Art des Festtages mit anzugeben, beispielsweise in folgender Form:

```
Claudias Geburtstag (Birthday)
```

Abschließend bietet Ihnen das Programm an, diesen Termin abzuspeichern. Haben Sie die Taste [Y] für "yes" gedrückt, wird abgespeichert, und Sie können weitere Termine eingeben oder aber mit der Taste [Esc] zum Hauptmenü zurückkehren.
Über den Menüpunkt "3. View Events" können Sie sich Ihre bisher eingetragenen Termine nacheinander anzeigen lassen, Veränderungen an ihrem Datum, ihrer Vorwarnzeit und ihrer Beschreibung vornehmen oder sie ganz einfach wieder löschen. Wenn Sie sich dann wieder im Hauptmenü befinden, können Sie mit der Taste [P] eine Liste der Termine auf Ihrem Drucker ausgeben lassen.
»Badger« weist bereits eine Anzahl von vorgegebenen Terminen auf, die selbstverständlich auch gelöscht werden können. Sie sind zum Teil auf amerikanische Verhältnisse zugeschnitten (wie Thanksgiving und der Unabhängigkeitstag), teilweise aber auch in Deutschland recht nützlich (wie Muttertag und Vatertag, Beginn der Jahreszeiten, Valentinstag).

Automatischer Start

Sinnvollerweise sollte »Badger« in Ihrer »startup-sequence« aufgerufen werden, so daß es bei jedem Booten Ihres Amiga automatisch gestartet wird. Denn auch das beste Erinnnerungsprogramm nützt Ihnen nichts, wenn Sie vergessen, es zu verwen-

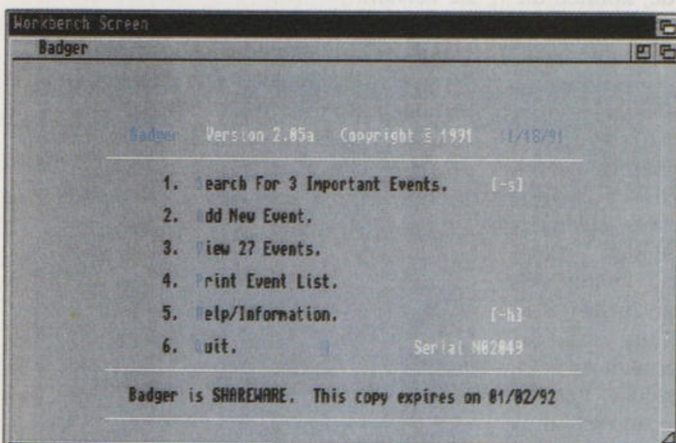


Bild 1. »Badger« entlastet Ihr Gedächtnis

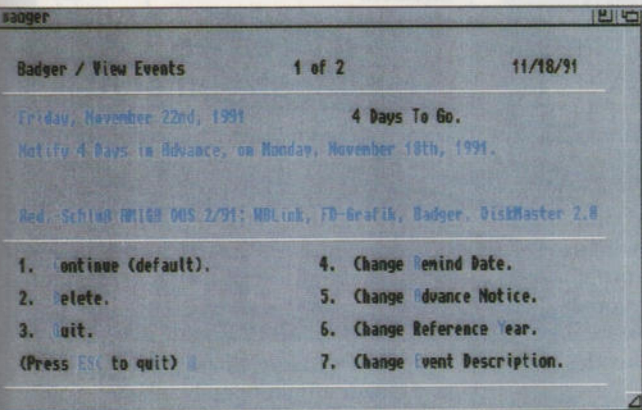


Bild 2. Schluß mit vergessenen Hochzeitstagen und Abgabeterminen

den! Wenn Sie beim Programmaufruf die Option "-s" angeben, macht sich »Badger« nur dann bemerkbar, wenn Termine "fällig" sind, wenn der aktuelle Tag also innerhalb ihrer Vorwarnzeit liegt. In diesem Fall werden der Reihe nach alle diese Termine angezeigt. Dies entspricht dem Menüpunkt "1. Search For X Important Event".

Die Option "-o" hat prinzipiell dieselbe Auswirkung, allerdings führt sie dazu, daß die Termine nur einmal pro Tag (beim ersten Boot-Vorgang) überprüft und gegebenenfalls angezeigt werden.

Falls das Programm "fällige" Termine entdeckt, gibt es den Return-Code 5 an das CLI zurück. Dies können Sie ausnutzen, um dann eine Meldung ausgeben zu lassen, wenn keine Termine vorhanden sind. Fügen Sie zu diesem Zweck folgende Zeilen in Ihre »startup-sequence« ein:

```
badger -s
if not warn
echo "Heute keine Termine. Mach Dir einen schönen Tag!"
endif
```

»Badger« ist Shareware. Es darf 30 Tage lang getestet werden. 45 Tage nach dem erstmaligen Start löscht es seine Daten-Files und kann nicht mehr verwendet werden. Für 15 US-Dollar (knapp 30 DM) erhalten Sie vom Programmautor eine registrierte Version ohne diese Einschränkung. Auch eine MS-DOS-Version des Programms ist für zusätzliche zehn Dollar erhältlich.

Ob Ihnen ein einfaches, aber wirkungsvolles Erinnerungsprogramm so viel wert ist, müssen Sie selbst entscheiden. Die Testperiode von immerhin mehr als sechs Wochen gibt Ihnen ausreichend Zeit, um festzustellen, ob das Programm für Ihre Anforderungen das Richtige ist. ❑

Hartmut Schumacher (vb)

Info:
Name: Badger
Autor: George Kerber
Merkmal: Shareware, Gebühr 15 US-Dollar
Quelle: Fish-Disk 543

Betrifft Shareware

Das Bezahlen von Shareware-Gebühren scheint immer noch im argen zu liegen. Man muß sich allerdings darüber im klaren sein, daß man mit Raubkopien arbeitet, solange man die Gebühren nicht zahlt. Deshalb gibt es seitens der Autoren auch so drastische Sicherungsmaßnahmen wie in »Badger«. Also, ehrlich bleiben und, falls benötigt, mal den eigenen PD-Händler fragen.

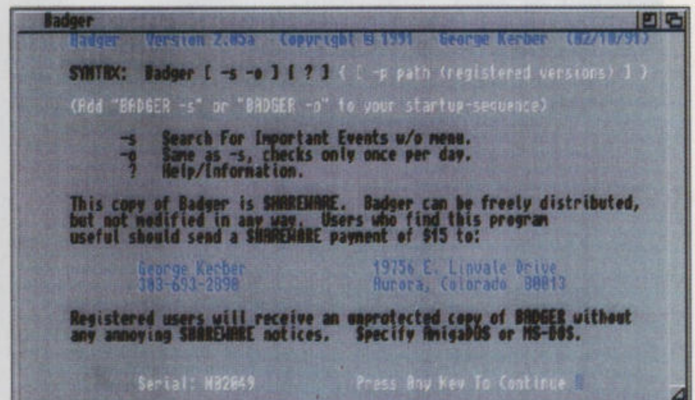


Bild 3. Syntax von »Badger« erläutert

FD-Serie

ABCD-Reihe, kurz belichtet

Vorbei ist die Basic-lose Zeit. Für BASIC-Freunde dieser Programmiersprache gibt es seit geraumer Zeit eine eigene Public-Domain-Serie.

Vorbei scheint die Zeit, als BASIC-Programme auf Public-Domain-Disketten eine Seltenheit waren. Stolze zwölf Monatsdisketten hat der Amiga-BASIC-Club-Deutschland bisher herausgebracht. Abwechslungsreich ist der jeweilige Inhalt der Disketten. Das Spektrum reicht von kniffligen Knobelspielchen bis zu ausgereiften Anwendungsprogrammen. Bei fast allen Programmen liegt der Sourcecode als BASIC-Datei vor.

Manche Programme wurden auch kompiliert und machen so einen recht flotten Eindruck. Eins jedoch vorweg:

Aus Copyright-Gründen muß AmigaBASIC vom Anwender selbst draufkopiert werden. Die Disketten sind selbststartend und melden sich meist mit einem kurzen Intro. Kommen wir nun zum Inhalt der einzelnen Disketten.

Fünf verschiedene Demos sind auf der Diskette #1 zu finden. Dabei reicht die Palette von 64er-Emulationen bis zum großen Scroller mit Sprites und allerlei Bewegung auf dem Bildschirm. Erstaunlich, was man so alles mit den Systemroutinen (Libraries) auf die Beine stellen kann.

Auf dieser Diskette befindet sich noch eine gepackte und kompilierte Version von »Memory« (die nochmals auf Disk #8 enthalten ist) und eine aufgepeppte Version (mehr Levels, Soundtrackersounds, Titelbild und schneller) von »Stoneage« (auch auf Disk #5). Der Sourcecode der Intros sind auf Anfragen beim Club erhältlich.

Spiele, Demos und noch mehr

Die zweite Diskette bietet ein actiongeladenes Text-Adventure namens »Sword« mit recht gut animierter Einführungsgrafik. Wer gerne Führungsqualitäten unter Beweis stellen möchte, der sollte »Valley of the Azteks« spielen. In diesem Spiel geht es um Krieger, Korn, Länder und Macht. Schlägt der Versuch fehl, so wird man vom Volk oder den Götter-Priestern enthauptet. Zwei Demos, die eine Animation der Fernseh-ARD-Eins zeigen, komplettieren diese Disk. Auch hier ist der Sourcecodes der Intros beim Club erhältlich.

Ein BASIC-Compiler ist immer etwas Besonderes. Auf der dritten Monatsdiskette ist der BASIC-Compiler »Cursor« in der Version 1.2 enthalten (siehe AMIGA DOS 2'91, Seite 77). Das Programm ist voll menügesteuert, so daß man über die Funktionstasten schnell zwischen Editor, AmigaBASIC und Compiler hin und her schalten kann. Dieser Compiler ist jedem BASIC-Programmierer zu empfehlen.

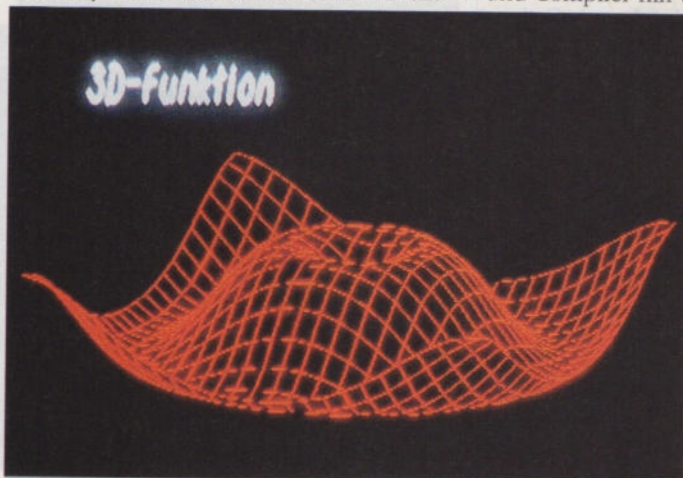


Bild 1. Mathematisches mit »3D-Funktion«

Disk Nummer #4 besteht aus zwei Disketten. Es handelt sich um das Handelsspiel »Trader« (siehe AMIGA DOS 10'91, Seite 140) und besticht durch gute Grafik und Spielspaß auch über längere Zeit.

Im Zeichen der Spiele steht auch Disk #5. Galaktisch geht es bei dem Wurmspiel »Galaktic-Wurm« zwar nicht zu, aber Schnelligkeit und Reaktion sind gefragt, denn sonst beißt man sich in sein eigenes, rasch anwachsendes Hinterteil.

Einsammeln und Steine verrücken ist das vorrangige Ziel beim Boulder-Dash-Komplizen »Stoneage«. Das Anfangs-Level ist noch recht einfach, aber dann arbeiten die Hirnzellen schon auf Hochtouren, um die weiteren Levels lösen zu können. »Amiga-Wurm« ist eine Tron-Variante für zwei Mitspieler.

Wie in Las-Vegas geht es bei »MCPoker« zu. Die Grafik des Spiels ist sehenswert und steht teureren Poker-Spielchen in nichts nach (bis zu vier Mitspieler sind möglich). Drei weitere Spiele vervollständigen diese Disk: »Slider«, ein nett aufbereitetes Verschiebespielchen, Bekanntes wie das Brettspiel »Shogun« und schließlich »Othello«, ein weiteres Denk- und Schiebespiel, mit guter Grafik und toll umgesetzten 3D-Effekten.

Lernen einfach gemacht

Lernen steht im Vordergrund der sechsten Disk. Von einem sechsteiligen, recht gut lesbaren Einsteiger-BASIC-Kurs geht es weiter zu einigen Programmen wie »Funktionsplot« und »3D-Funktion«, zwei Programme, die sich sehr ausführlich mit Kurvendiskussion befassen. »Amiga-Calc« ist ein einfacher Taschenrechner, der neben den vier Grundrechenarten auch Wurzel- und Quadratfunktionen beinhaltet. Solche Programme können gerade für Schüler nützlich sein.

Eine recht komfortable Adreß- und Datenverwaltung ist mit »Adress 500« auf Disk #7 enthalten. Das Programm bietet einige Features, wie beispielsweise unterschiedlich formatierter Ausdruck, editierbarer Druckertreiber, um nur zwei zu nennen. Die Benutzeroberfläche des Programms ist grafisch gut gelöst. Die Kriterien für die Eingabe können beliebig verändert werden. Die Eingabe selbst erfolgt über einen komfortablen Zeileneditor und nicht über die normale Input-Anweisung von AmigaBASIC.

Anwendungen und Tools

Für Kurzweil sorgt Disk #8: »Imperium« ist eine Handelssimulation, bei der es um Land, Hotels, Restaurants und Spielcasinos geht. Zwei bis sechs Spieler können versuchen, die Alleinherrschaft über 58 Ländereien in Europa, Asien und Afrika zu erhalten.

»Memory« nennt sich eine Umsetzung des gleichnamigen Kartenspiels, das bereits jeder schon einmal gespielt haben dürfte. In dieser Version ist der Sourcecode von Disk #1 enthalten, sie läuft jedoch bei weitem nicht so schnell.

Vielfalt ist Trumpf

Bunt gemischt präsentiert sich die Programmpalette auf Disk #9. »Cellur Automata« ist eine Umsetzung der bekannten Life-Simulation. »Life« simuliert eine Zellenkolonie, die, durch Wachstum und Zerstörung beeinflußt, überleben muß. Besonders für Spieleprogrammierer dürfte das Programm »IffBrush2BoB« interessant sein. Es wandelt IFF-Brushes (zum Beispiel von »DPaint«) in AmigaBASIC-kompatible Object-Dateien.

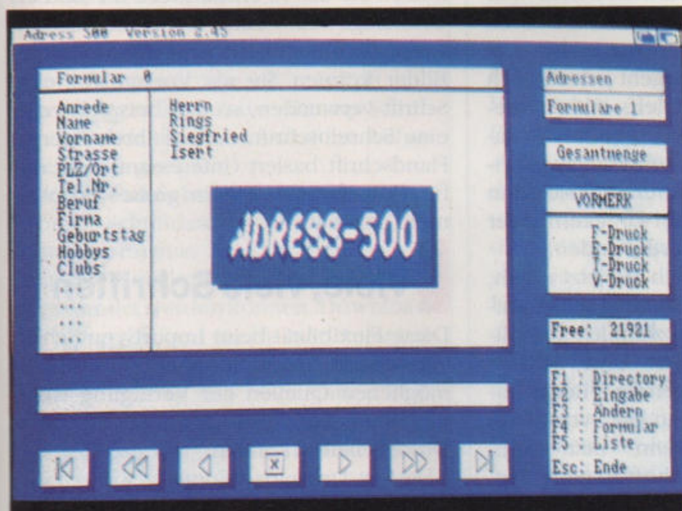


Bild 2. Verwalten mit »Adress 500«

»BmapReader« entschlüsselt die unlesbaren ».bmap-Dateien« und gibt Aufschluß über deren Inhalt. »Superterm« ist ein DFÜ-Programm, das über Xmodem-Protokolle und volle Menüoberfläche verfügt. Start-Stop-Parity-Bits und Übertragungsgeschwindigkeiten können eingestellt werden.

Musikalisch

Das Programm »Synthesizer« lockt seltene Klänge aus dem Lautsprecher. Und schließlich zaubert »Worldmap« die Weltkarte auf den Screen.

Fun-Programme verspricht Disk #10. Bei »AGEtest« werden Sie mit Fragen zu Ihrer Person förmlich "bombardiert". »WitchingHour« ist ein sehr kniffliges Verschiebespiel. Mit »Vampire« stellt sich ein kleines Text-Adventure vor (leider nur in Englisch). Komplexer geht's bei »Wizard« zu (ebenfalls ein Text-Adventure).

Falls Sie vergeßlich sind, ist »Geburtstagskalender« das Richtige für Sie und Ihre Jubilare. Der absolute Renner dieser Disk ist jedoch »Monopoly«, dem gleichnamigen Gesellschaftsspiel nachempfunden.

Vollgepackt mit Beispielprogrammen, stellt sich Disk #11 vor: Neben einem recht komfortablen Datei-Requester (öffnet ein eigenes Window zur Auswahl von Dateien) und einer Super-Dateiverwaltung befindet sich auf der Diskette ein Programm zum Lesen von Texten (ähnlich dem FD-Textanzeigeprogramm »PPmore«). Die drei Zeichenprogramme (»Amigapaint«, »Draw256« und »Malen«) wären eigentlich schon eine eigene Diskette wert.

Zwar ist der Stern »DPaint« noch weit entfernt, aber für den Hobbymaler sind diese Programme allemal hilfreich. Mit dem kleinen Programm »Stars« kann man die Positionen der intergalaktischen Sternenwelt berechnen. »Fractal-Mountain« erzeugt fraktale Landschaften auf dem Monitor. Diese sogenannten Fraktale beruhen auf ganz speziellen, geometrischen Berechnungen und erzeugen unregelmäßig geformte Objekte und Oberflächen. Das kleine Programm »Grids« setzt Wellen und Kurven in eine hörbare Akustik um. Ebenfalls mit Kurven beschäftigt sich »Mathgraphic«. Zu guter Letzt komplettiert das Spiel »Enigma« diese Diskette.

Diskette #12 stellt insgesamt vier Anwenderprogramme vor.

Mit »M.R.A.« können Sie Mahnungen, Rechnungen und Angebote schreiben. Schließlich gibt es noch das Verwaltungsprogramm »UniversalDatei«, das zwar in Bedienung und Oberfläche sehr einfach gehalten ist, dafür aber sehr übersichtlich und leicht an eigene Gegebenheiten anzupassen ist.

So wurde bei diesem Programm beispielsweise auch an Import- und Export-Funktionen für Formulare gedacht. Mit »ADV« und »AVSBDP« befinden sich schließlich noch zwei weitere Adreßverwaltungen auf dieser Diskette.



Bild 3. Grübeln mit »Othello«

Abschließende Bemerkung

Alles in allem stellt diese Serie eine Fundgrube für BASIC-Freunde dar. Der Bezugspreis für eine Diskette ist mit 4 Mark noch relativ niedrig, wenn man bedenkt, daß findige Händler alleine für ein Programm wie »Amigapaint« bereits 12 Mark und mehr abkassieren.

Wer seine eigenen BASIC-Programme für den BASIC-Club einschickt, bekommt die Monatsdisketten fast kostenlos (lediglich Rückporto und Leerdisketten). Clubgebühren werden nicht extra erhoben. Jeder, der eigene Programme einschickt, wird automatisch Mitglied. Es lohnt sich auf jeden Fall. ❑

Siegfried Rings (vb)

Info
Amiga-BASIC-Club Deutschland
c/o Gerd Schmitt
Kurt-Schumacher-Str. 54
5470 Andernach
(Rückporto nicht vergessen)

Tool

Typographer

Ein Shareware-Programm sorgt für Schriftenvielfalt bei 24-Nadel-Druckern und Tintenstrahldruckern.

Schriftenvielfalt ist auf dem Amiga prinzipiell kein Problem: Die meisten Textverarbeitungsprogramme bieten die Möglichkeit, mehrere Schriften zu verwenden. Die Ausdruckqualität ist dabei allerdings meist nur mangelhaft bis mäßig.

Desktop-Publishing-Programme hingegen können selbst mit Nadeldruckern Schriften in annehmbarer Qualität ausgeben. Das (vor dem Ausdrucken nötige) Berechnen einer Seite mit mehreren unterschiedlichen Schriften dauert jedoch recht lange.

Der Einsatz einer Vielzahl von Schriften in wirklich ansprechender Druckqualität ist ohne lange Berechnungszeiten also nur dann möglich, wenn man auf die internen Schriften des jeweiligen Druckers zurückgreift. Auf diese Weise können Texte in der besten dem Drucker möglichen Qualität ausgegeben werden – eine unerläßliche Voraussetzung für vorzeigbare Dokumente.

Diese Verfahrensweise ist in aller Regel jedoch recht unbefriedigend, denn oft benötigt man mehr oder ganz einfach andere Schriften als die, die im Drucker bereits eingebaut sind.

Die Lösung für dieses Problem liegt in der Tatsache, daß sowohl Laserdrucker als auch die (wesentlich preisgünstigeren) Tintenstrahl- und Nadeldrucker sogenannte *Download-Schriften* in ihren Speicher laden können. Diese Schriften lassen sich dann vom Benutzer auf dieselbe Weise benutzen wie die eingebauten Schriften – und auch ihre Druckqualität steht der der internen Schriften in nichts nach.

"Fantastisch", werden Sie sagen, (sofern Sie einen der vielen Download-fähigen Drucker besitzen), "aber wie kommt man an solche Download-Schriften heran?" Für Laser- und Tintenstrahldrucker existieren Schriftdisketten, die man für relativ viel Geld erwerben kann. Es geht jedoch auch preiswerter: Dietmar Eilerts Shareware-Programm »Typographer« (von der Antares-Disk 82; siehe Bild 1) ermöglicht es, eigene Schriften zu entwerfen und sie dann problemlos mit 24-Nadel-Druckern oder Deskjet-Kompatiblen zu verwenden.

"Schön und gut", höre ich Sie jetzt sagen, "aber Schriften von Grund auf selbst entwerfen? Das klingt doch ziemlich kompliziert und zeitaufwendig." Recht haben Sie, vor allem wenn man kein Schrift-Experte ist. Aber das ist auch gar nicht nötig:

Der »Typographer« nimmt Ihnen diese Arbeit ab.

Denn wenn man es sich recht überlegt, gibt es ja für den Amiga bereits eine Unzahl von Schriften – auch wenn sie im nicht sehr praktischen Bitmap-Format vorliegen: Bereits im Lieferumfang des Amiga sind eine ganze Reihe solcher *Bitmap-Schriften* enthalten, und auch auf unzähligen FD-Disks finden Sie weitere Schriften: große, kleine, moderne, altertümliche, einfache, verschnörkelte; praktisch alles, was Ihr Herz begehren kann – bis hin zu kyrillischen (Fish-Disk 427, Fish-Disk 517) und griechischen (Fish-Disk 474) Schriften.

Nun wäre es doch sehr schade, wenn diese Vielfalt ungenutzt brachliegen würde. Folgerichtig können Sie mit dem »Typographer« eine Bitmap-Schrift laden und als Download-Schrift wieder abspeichern.

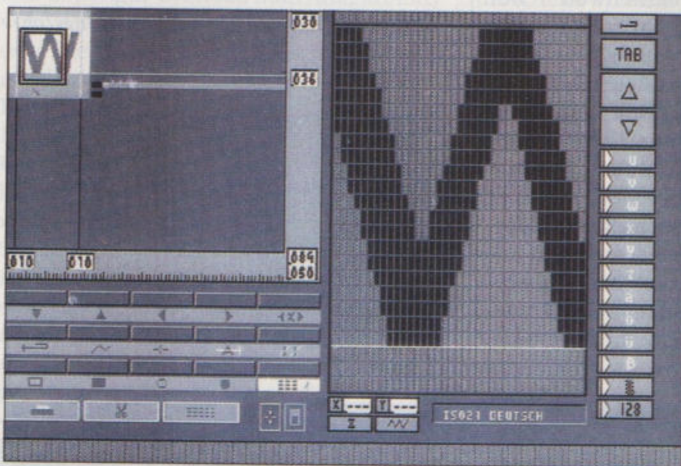


Bild 1. »Typographer«: ein flexibler Schrifteneditor

Und schon hat Ihr Drucker eine Schrift mehr – ohne daß Sie sie selbst hätten entwerfen müssen. Aber auch Download-Schriften, die für Laserjet-kompatible Drucker gedacht sind, können mit »Typographer« in Formate umgewandelt werden, die sich für Deskjet-Kompatible und 24-Nadler eignen. Und selbst *IFF*-Bilder kann Dietmar Eilerts Programm laden. Somit können Sie alle Schriftzeichen, die irgendein Programm auf den Bildschirm Ihres Amiga bringt, selbst verwenden (aus urheberrechtlichen Gründen oft jedoch nur für private Zwecke). Speichern Sie (mit einem Hilfsprogramm wie Roger Fischlins »Fenster« [Fish-Disk 362]) den Bildschirm des entsprechenden Programms ab, und benutzen Sie das so entstandene IFF-Bild als Grundlage für eine Download-Schrift. Auch mit einem Scanner hergestellte IFF-Bilder können Sie als Vorlage für eine Schrift verwenden, so daß beispielsweise eine Schreibschrift, die auf Ihrer eigenen Handschrift basiert (interessant nicht nur für Unterschriften), kein großes Problem mehr darstellt.

Viele, viele Schriften

Diese Flexibilität beim Import von Schriften bewirkt, daß Ihnen eine Vielzahl von möglichen Quellen zur Verfügung steht, die Ihnen Download-Schriften für Ihren Drucker liefern können:

- Deskjet-Download-Fonts;
- Laserjet-Download-Fonts;
- jeder Bitmap-Zeichensatz;
- jeder andere Font-Editor, der das Bitmap-Format unterstützt (beispielsweise der »FED« von der 1.3-Extras-Disk);
- jedes IFF-Zeichenprogramm (meist natürlich »DPaint«, ebensogut eignen sich jedoch die FD-Programme »Graffiti« [Fish-Disk 531] und »FreePaint« [Fish-Disk 548]).

Doch selbstverständlich erschöpft sich das Leistungsangebot des »Typographers« nicht nur darin, Schriften verschiedener Formate zu konvertieren: Sie können Schriften jeder der eben genannten Quellen auch verändern, bevor Sie sie im Format Ihres Druckers abspeichern. Oder Sie verwenden eine solche Schrift lediglich als Grundlage für eine eigene Schrift (interessant für Sonderzeichen wie mathematische Symbole, griechische Buchstaben oder phonetische Zeichen). Ganz Mutige, die zumindest entsprechende Grundkenntnisse besitzen, können sogar eine vollkommen neue Schrift selbst entwerfen.

Der »Typographer« kann eine geladene Schrift auch als IFF-Bild abspeichern (siehe Bild 2). Sie könnten also beispielsweise eine Bitmap-Schrift laden, sie als IFF-Bild abspeichern, dann mit Ihrem Lieblingsgrafik-

programm Veränderungen an den Buchstaben vornehmen und das so entstandene neue IFF-Bild wieder in den »Typographer« laden, um die Schrift dann schließlich als Download-Schrift abzuspeichern.
Sie sind jedoch nicht unbedingt auf weitere unterstützende Programme angewiesen. Denn auch der »Typographer« bietet Ihnen eine Reihe von Zeichenfunktionen, mit denen Sie einen geladenen Zeichensatz verändern können. Diese Zeichenoperationen finden in der vergrößerten Lupendarstellung auf der rechten Bildschirmseite statt. Neben den grundlegenden Zeichenfunktionen wie Freihand, Linie, Rechteck, Ellipse finden sich auch speziellere Funktionen wie Fettschrift, Umrandung, Kursivschrift, Invertieren und ein netter 3-D-Effekt. Eine *Undo*-Funktion ist dankenswerterweise ebenfalls vorhanden. Diverse Hilfs- und Begrenzungslinien erleichtern das Entwerfen von Buchstaben. Die Benutzeroberfläche des Programms ist recht ungewöhnlich: *Pull-down-Menüs* sind nicht vorhanden, statt dessen erfolgt die Bedienung über *Gadgets* und *Pop-up-Menüs*. Die rechte Maustaste dient dazu, Eingaben zu bestätigen oder aber eine Auswahl abzubrechen.
Das Bedienungsprinzip ist also durchaus durchdacht und nach einer kurzen Eingewöhnungszeit leicht zu beherrschen. Dennoch bleibt die Frage, wie sinnvoll es ist, ein eigenes (auch noch so schmackhaftes) Süppchen zu kochen, wenn doch alle Bemühungen der Software-Industrie dahin gehen, den Anwendern möglichst einheitliche Benutzeroberflächen zu bieten, um die Eingewöhnungszeit bei neuen Programmen auf ein Minimum zu reduzieren.
Um eine möglichst große Arbeitsfläche bieten zu können, öffnet »Typographer« einen Bildschirm im *Overscan*-Modus; die Bildlage läßt sich programmtechnisch einstellen. Das Programm benötigt einen Arbeitsspeicher von mindestens ein MByte. Große Schriften können jedoch mehr Speicher erfordern. Ein Deskjet-Drucker muß mit einer 128-kByte oder 256-KByte-Speicherkarte ausgerüstet sein. Damit sind Sie dann sehr viel flexibler, als wenn Sie fertige Schriftmodule für diesen Drucker erwerben würden.
Die Shareware-Gebühr für »Typographer« beträgt 30 DM. Dies ist wohl kaum zuviel verlangt für ein derart flexibles, durchdachtes und vor allem leistungsfähiges Programm.
Die unregistrierte Programmversion auf der Antares-Disk 82 besitzt nur eingeschränkte Download- und Speicherfunktionen und wird in unregelmäßigen Zeitabständen für einige Sekunden durch einen Requester lahmgelegt, der den Benutzer an die Zahlung der Shareware-Gebühr erinnert. Registrierte Anwender erhalten neben der voll funktionsfähigen Programmversion eine gebundene und bebilderte Anleitung. ❑

Hartmut Schumacher (vb)

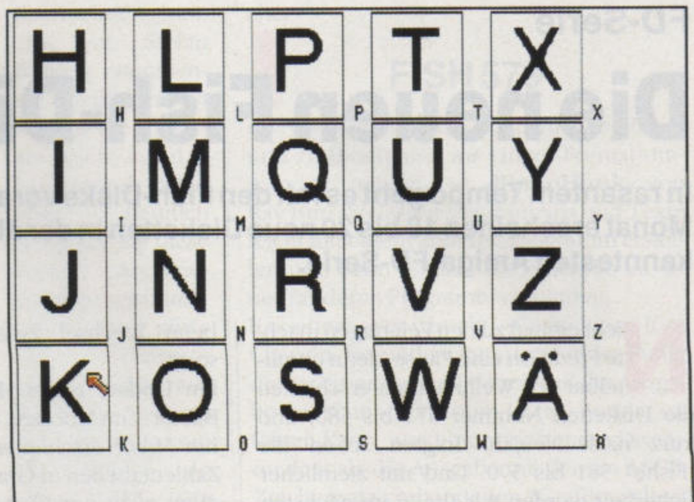


Bild 2. »Typographer« kann Schriften als IFF-Bilder abspeichern

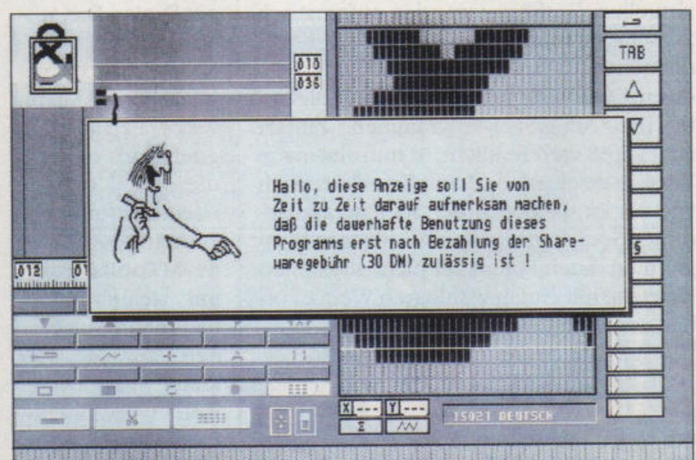


Bild 3. Ein Requester mahnt die Shareware-Gebühr an

Glossar

Hier werden alle *kursiv* gesetzten Wörter des Artikels erklärt.
Bitmap-Schriften: Schriften, die in bestimmten, festen Größen vorliegen. Vergrößert man eine solche Schrift, entstehen gezackte Ränder. Bitmap-Schriften sind daher für ernsthaftere Aufgaben kaum geeignet.
Download-Schriften, (oder Softfonts): Schriftdateien, die in den Speicher bestimmter Drucker geladen und dann wie die internen Druckerschriften verwendet werden können. Download-Schriften bleiben nur bis zum Ausschalten des Druckers in dessen Speicher erhalten, müssen also beim Starten Ihres Systems jedesmal neu von Diskette (oder Festplatte) in den Drucker geladen werden.
Gadgets: Symbole, die der Benutzer mit der Maus anklicken kann, um Programmfunktionen aufzurufen.
IFF: Abkürzung für "Interchange File Format". Dies ist ein Oberbegriff für eine Reihe von Standard-Formaten, die den Datenaustausch (Grafiken, Animationen, Samples, Texte) zwischen verschiedenen Programmen ermöglichen.
Overscan: Ein Bildschirm-Modus, in dem auch die normalerweise ungenutzten Randbereiche Ihres Monitors verwendet werden.
Pop-up-Menüs: Auswahlleisten, die unter der jeweiligen Position des Mauszeigers erscheinen, wenn Sie aufgerufen werden, und es dadurch dem Benutzer ersparen, den Mauszeiger an den oberen Bildrand zu bewegen, um einen Menüpunkt anzuwählen.
Pull-down-Menüs: Auswahlleisten, die vom oberen Bildschirmrand herunterklappen, wenn Sie mit der (meist rechten) Maustaste aufgerufen werden. Bestes Beispiel: die Workbench-Menüs.
Undo: Macht die zuletzt angewählte (Zeichen-)Operation wieder rückgängig.

Info
Name: Typographer
Version: 2.0
Autor: Dietmar Eilert
Quelle: Antares 82

FD-Serie

Die neuen Fish-Disks

In rasantem Tempo geht es mit den Fish-Disks voran: Monat für Monat erscheinen 10 bis 20 neue Disketten in der ältesten und bekanntesten Amiga-FD-Serie.

Nicht einmal zu den Feiertagen machte Fred Fish eine Pause, denn unmittelbar vor Weihnachten erschienen die Disketten Nummer 571 bis 580, und kurz nach Neujahr folgten schon die "Fishe" 581 bis 590. Und mit ziemlicher Sicherheit werden wir Ihnen im nächsten Monat schon das nächste Jubiläum vorstellen können: Bis dahin sollten die Fish-Disketten bis 600 fertiggestellt sein.

FISH 571

571 – Von Nic Wilson sind wieder einmal zwei Updates dabei: das DMouse-ähnliche Tool **»AutoCLI«** und der Speedtester **»SysInfo«**, das nun schon bei Version 2.53 angelangt ist. Neu ist jedoch **»CPUSet«** vom selben Autor, das für 68020/030/040-Prozessoren geschrieben wurde und das bequeme Ein-und Ausschalten von Burst-Modi und Caches (Pufferspeicher innerhalb des Prozessors) per Mausklick erlaubt.
Howard Andersons **»Gwin«**, eine Grafik-Bibliothek, die sich bisher nur an C-Programmierer richtete, ist in der neuen Version 2.0 zu einer "shared Library" geworden, die nun von beliebigen Programmen aufgerufen werden kann. Darin sind viele Funktionen zum Erzeugen von Screens, Menüs und Requestern enthalten, die Programmierern, die noch unter 1.3 ihre Programme entwickeln, sicherlich nützlich sein können. Dazu kommen noch Funktionen zum Umgang mit dem eingebauten virtuellen Grafiksystem, das automatisch zwischen tatsächlichen und Screen-Koordinaten umrechnet.
»MemClear« ist schon ein echter Oldie unter den Amiga-FD-Programmen. Zuletzt auf Fish 58 veröffentlicht, ist nun eine neue Version erschienen, deren Zweck es nach wie vor ist, Fehler in der Speicherallozierung (Speicherzuordnung) eigener Programme durch Füllen der nicht allozierten Bereiche mit einem wählbaren Wert zu offenbaren.

FISH 572

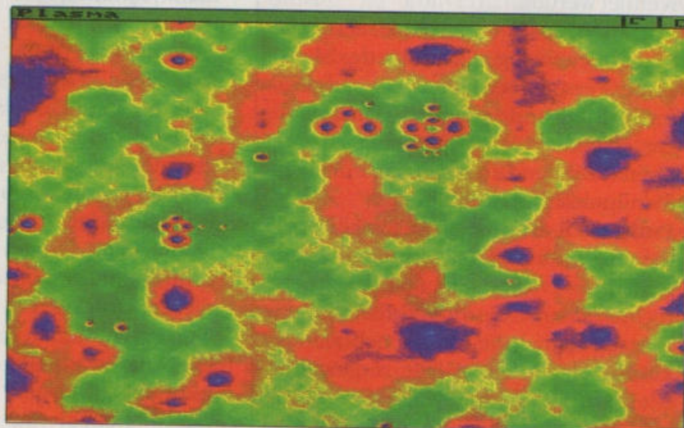
572 – Vom selben Autor stammt auch **»CrossFade«**, ein kleiner Screenhack, der für einen besonders weichen Übergang beim Wechsel zwischen zwei Screens sorgt.
Ein Update ist zu **»Multiplot«** von Alan Baxter, Tim Mooney, Rich Champeaux und Jim Miller erschienen. Die Software, die Zahlentabellen in Grafiken umsetzt, unterstützt nicht nur Plotter zur Ausgabe, sondern auch Postscript-Drucker sowie HP-Laserjet-Drucker. Mit dem mitgelieferten Device »PLT:« lassen sich aber auch alle anderen Preferences-Printer zur Ausgabe nutzen, wenn auch mit einer gewissen Wartezeit für die Berechnung der Bitmap der zu zeichnenden Grafiken.
Toolmanager-Programmierer Stefan Becker hat mit **»ResetHandler«** und **»WBStart«** zwei neue Programme herausgebracht. Ersteres installiert einen Reset-Handler, der allen, die über einen geeigneten Amiga verfügen, vor einem Reset noch 10 Sekunden Bedenkzeit gibt. "Geeigneter Amiga" bedeutet in diesem Zusammenhang einen Amiga mit gepuffertem Reset, was leider alle Amiga 500 ausschließt: Hier geht die Reset-Leitung direkt von der Tastatur zum Prozessor.
»WBStart«, das andere der beiden Programme, emuliert für Programme, die es benötigen, einen Start von der Workbench, obwohl aus der Shell heraus aufgerufen.

FISH 573

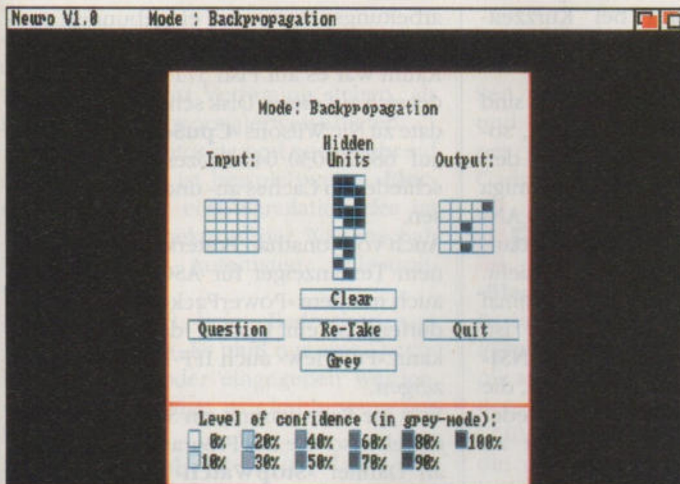
573 – Sie bietet als erstes noch ein Plot-Programm, diesmal **»MathPlot«** von Rüdiger Dreier. Das ausschließlich für Kickstart 2.0 geschriebene Shareware-Programm greift auf die aus diesem Grunde ebenfalls auf der Diskette vorhandene **»MToolLibrary«** mit vielen mathematischen Funktionen zurück. Der kleine Freeware-Bruder von »MathPlot« namens **»Plotter«** findet sich auch auf dieser Disk. Von Bryan Ford stammt der **»MultiPlayer«**, ein Abspielprogramm für die Musikmodule von mehr als 15 Musikprogrammen, darunter auch »Soundtracker«, »Noisetracker« und »MED«. Besonderer Gag: Unter 2.0 ist das Fenster des Multiplayers ein Application-Window – Icons von Musikstücken können also einfach auf das Fenster abgelegt werden, um abgespielt zu werden. Die Tryware-Version bietet dabei keinen ARexx-Port und keine Möglichkeit, programmierte Abfolgen zu laden oder zu speichern.
Hübsche Bilder erzeugt **»Plasma«** von Roger Uzun, das auch in einer Version für Amigas mit FPU (Fließkomma-Arithmetik-Prozessor) vorliegt. Leider wurde eine Routine zum Abspeichern vergessen, aber dafür gibt's zum Glück **»ScreenX«** und ähnliches.

FISH 574

574 – Auch **»Chemesthetics«** von Jörg Fenin ist überarbeitet worden und liegt nun in der Version 2.10 vor. Das Programm stellt Moleküle unter Verwendung des Kalottenmodells grafisch auf dem Bildschirm dar.
Michael Sinz hat die neueste Version des bekannten Festplatten-Speedtesters **»DiskSpeed«** sowie die Software-Lupe **»MKSLens«** beigesteuert. »DiskSpeed« mißt in der Version 4.1 nun nicht mehr nur den reinen Datendurchsatz, sondern auch den Anteil der CPU-Zeit, den die Treibersoftware der Festplatte dabei für sich beansprucht. »MKSLens« stellt ständig in einem eigenen Fenster den Screen-Bereich um den Mauszeiger herum vergrößert dar, was bei Detailarbeiten durchaus hilfreich sein kann.
»NewZAP«, wiederum von Dallas J. Hodgson, ist in der neuen Version 3.3 endlich kompatibel zum OS 2.0. Besonderes Feature dieses Disk-Editors ist, daß er dank



»Plasma« von Fish 573



»Neuro« erkennt Informationsmuster

eines speziellen, schmalen Fonts auch auf einem NTSC-Screen alle 512 Bytes eines Sektors gleichzeitig darstellen kann.

FISH 575

575 – Die Auswahl in Sachen Plot-Programme bereichert auch **»DataPlot«** von J. Dale Holt. Die Ausgabe ist auf Epson-kompatible Drucker direkt möglich oder auf dem Umweg über das Device »PLT:« auch auf jeden Preferences-Drucker.
Auch Nico Francois hat wieder einmal ein paar Programme beigesteuert. Neben Updates zur Entpack-Library **»PPLib«** und zur Requester-Library **»ReqTools«** ist auf Fish 575 auch **»AmigaToNTSC«** dabei. Dies ist ein Tool, das die »graphics.library« eines PAL-Amigas so patcht, daß andere Programme annehmen, es handele sich um einen NTSC-Amiga. Resultat: NTSC-Screens mit 60 Hertz Bildwiederholfrequenz.

FISH 576

576 – Von Andrew Hackett stammt der **»Analyzer«**, quasi ein Logik-Analyzer in Software, der jedoch naturgemäß nicht in echten Schaltungen, sondern nur anhand vorhandener Meßdaten eingesetzt werden kann.
Zu **»Budget«**, einem Finanzplaner von Le Lay Serge Camille, ist mit der Version 1.3.4 ein Update erschienen.
Das aus Frankreich stammende **»TermII«** von Eric Gontier hat mit dem in Deutschland viel bekannteren »Term« von Olaf Barthel nur rein namentliche Ähnlichkeiten, es entstand ansonsten völlig unabhängig. Doch auch bei »TermII« handelt es sich um ein recht komfortables Terminal-Programm, das ebenfalls auf OS 2.0 angewiesen ist.
577 – Auch an **»LhA«** werden in erster Linie die DFÜ-Fans ihre Freude haben. Das von Stefan Boberg stammende Programm ist sowohl zu »LhArc« als auch zu »LZ« kompatibel, dabei aber deutlich schneller als diese beiden Archivierungsprogramme. Obendrein gibt's ein Hilfsprogramm, das aus LhA-Archiven selbstextrahierende Archive erzeugt, die sich beim Start entpacken, ohne daß »LhA« selbst benötigt wird.

FISH 577

»OwnDevUnit« ist eine von Christopher Wichura stammende Library, die das leidige Problem löst, daß im AmigaOS keine Prioritäten für Zugriffe auf Ports festgelegt wurden. »OwnDevUnit« bindet hier einen erweiterten Schutzmechanismus ein, der es ermöglicht, einen Port von einem anderen Programm, das ihn gerade belegt, anzufordern.
Zu einem der älteren Freeware-Spiele, nämlich **»PetersQuest«** von David Meny, ist ebenfalls ein Update dabei. Eine Handvoll neuer Features macht das Spiel auch für Besitzer der älteren Version erneut interessant.
Ein interessantes Utility für Besitzer von Quantum-SCSI-Festplatten ist **»TurboQuantum«** von Ben Fuller, das durch Setzen eines speziellen Bits in der Platte dafür sorgt, daß diese sich nicht mehr während eines Transfers vom Bus abmeldet, und so in der Regel für Performance-Gewinne sorgt.

FISH 578

578 – Brett Larson machte sich die Mühe, den bereits einmal auf Fish 278 in Fortran veröffentlichten Circuit-Analyzer **»Spice3«** in C umzusetzen. Die auf dieser Disk vorhandene C-Version trägt die Versionsnummer 3.2. Voraussetzung ist mindestens 1 MByte Speicher.

FISH 579

579 – Ein nützliches Utility für alle, die ab und zu Binärdaten ins Linker-Format umwandeln müssen, ist **»BinToHunk«** von Ray Burr. So lassen sich Bitmaps und Texte leicht mit einem anderen Programm erstellen und dann einfach als Datenfile zu einem anderen Programm verbinden.
Kassettencover druckt **»Cass«** von Jörn Clausen. Besonders herausragender Punkt dieses Programms ist dabei, daß die Ausgabe auf Wunsch auch im LaTeX-Format erfolgen kann, was eine weit höhere Druckqualität als die Ausgabe mit den im ASCII-Zeichensatz vorhandenen "Grafikzeichen" erlaubt.
Bisher war er kommerziell, nun soll er als Demo für die neuere, nach wie vor kommerzielle Version 2.0 dienen: Der **»OctaMED«** 1.00b von Teijo Kinnunen, im Gegensatz zum bisherigen Freeware-MED achtstimmig. Im Gegensatz zu sonstigen Demos von kommerziellen Programmen sind in diesem Falle alle Funktionen vorhanden, es handelt sich lediglich nicht um die aktuellste Version.

FISH 580

580 – Was die letzte veröffentlichte Version von **»WorldDataBank«** noch vermissen ließ, bietet die neue Version 2.2: das detaillierteste, über ein MByte lange Datafile, das nur dank LhA-Archivierung auf die Diskette paßte. Jetzt können die beigefügten Programme zum Zeichnen der Weltkarte aus verschiedenen Perspektiven, also auf die volle Auflösung zurückgreifen.
581 – Der **»FileLocator«** von Joe Rattz hilft beim Wiederfinden von Files auf bis zu 28 verschiedenen Devices. Per Wildcard werden alle Verzeichnisse rekursiv durch-

»SysInfo« – neues Update



Brettspiel »Diplomacy« auf den Amiga umgesetzt

sucht, bis das File gefunden wurde. Neuronale Netze lassen sich mit **»Neuro«** von Berthold Ruf und Ulrich Wisser simulieren. Das simulierte Netz ist in der Lage, bestimmte Informationsmuster zu lernen und wiederzuerkennen, einige Beispiele werden ebenfalls mitgeliefert.

FISH 581

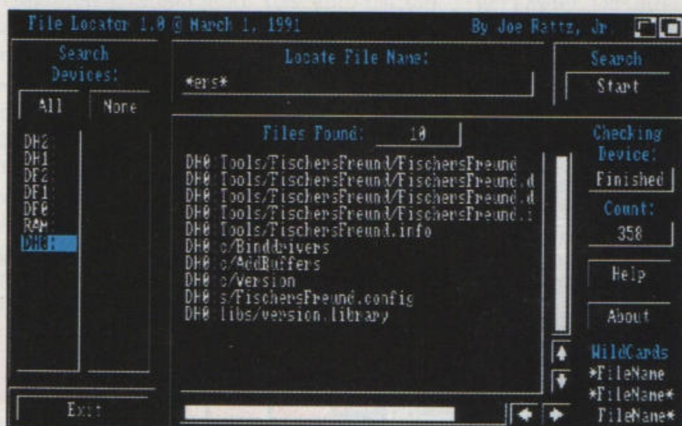
Eine Gemeinschaftsproduktion von Syd Bolton, Chris Timmerberg und Colin Vernon ist **»Prism«**. Dieser ANSI-Editor zeichnet sich vor allem dadurch aus, daß er auch die Erstellung von ANSI-Animationen erlaubt. Besonders beliebt ist diese Art von animierten Texten in Mailbox-Systemen.
Eine Stoppuhr simuliert **»StopWatch«** von Christian Danner. Die Funktion der Start-Stop-Taste übernimmt dabei der Feuerknopf des Joysticks, der natürlich mit einem selbstgebauten Anschlußkabel auch durch jede andere Art von Tastern ersetzt werden kann. Interessante Anwendungen ergeben sich zusammen mit einem Reedrelais bei Kurzzeitmessungen im Hobbybereich.
ANSI-Codes sind schön und gut, solange man sich den Text auf dem Amiga und über einen ANSI-fähigen Textanzeiger ansieht. Wenn das einmal nicht der Fall ist, hilft **»StripANSI«** von Syd Bolton, die Steuercodes wieder aus dem Text zu entfernen.

FISH 582

582 – Fish 582 bringt die Computerversion des Brettspieles **»Diplomacy«**, das Steve Douthat auf den Amiga umgesetzt hat. Der Computer ist dabei jedoch nicht in der Lage, mitzuspielen, sondern übernimmt nur die Verwaltungsarbeiten und die Darstellung des aktuellen Spielfeldes.
Zum Erstellen, Verwalten und Präsentieren von Multiple-Choice-Tests ist **»Questionatort«** von Erik von Raspator geschrieben worden.
Und für alle, die immer noch nicht über eine Datenbank ihrer Videofilme verfügen, bietet **»VideoDat«** von Sascha Fengel eine erneute Gelegenheit, ihre Programmsammlung um eines dieser nützlichen Hilfsprogramme zu ergänzen.

FISH 583

583 – Den wohl bisher vollständigsten und umfangreichsten "Lookalike" zum Unix-Dienstprogramm »nroff« liefert Dennis Gournelle mit seinem **»Aroff«**. Für Shell-gestützte Textformatierung immer noch eines der Feature-reichsten Programme.
Auch **»ATbl«** vom selben Autor orientiert sich an einem Unix-Vorbild, in diesem Falle dem Tabellen-Tool »tbl«. Möglich ist sowohl die Ausgabe auf einen Drucker mit IBM-Grafikzeichen als auch die Ausgabe einer IFF-ILBM-Grafikdatei, die problemlos in die gängigen Textverarbeitungsprogramme eingebunden werden kann.



»FileLocator« findet jedes File

Kaum war es auf Fish 571 erschienen, findet sich auf dieser Disk schon das erste Update zu Nic Wilsons **»CpuSet«**, mit dem sich auf 68020/030/040-Prozessoren die verschiedenen Caches an- und abschalten lassen.
Auch von Jonathan Potters **»FullView«**, einem Textanzeiger für ASCII-Dateien, die auch mit dem »PowerPacker« gepackt sein dürfen, ist ein Update dabei. Nebenbei kann »FullView« auch IFF-ILBM-Bilder anzeigen.
Von Joe Rattz stammt ein Stoppuhr-Ersatz, der ebenso wie das Programm von Christian Danner **»StopWatch«** getauft wurde. Verwechslungen scheinen da fast vorprogrammiert.
Weiter geht's mit zwei Programmen von Jonathan Potter. **»SuperLock«** ist ein Tool, mit dem sich beliebige Devices sperren lassen, so daß kein Zugriff mehr auf sie möglich ist. Optional können auch Tastatur und Maus gesperrt werden, was sinnvoll ist, wenn beispielsweise auf Computertreffs ein Rechner gegen unbefugte Benutzung gesichert werden soll.
Das andere "Super"-Programm ist **»SuperSpell«**, ein Rechtschreibkorrekturprogramm, das automatisch im Hintergrund jegliche Tastatureingaben überpüft, unabhängig davon, mit welchem Programm man arbeitet. Damit eignet es sich nicht nur für Textverarbeitungen, sondern auch für Editoren und sogar zur Korrektur von Texten, die man online in einer Mailbox schreibt. Dabei ist ein Wörterbuch von 9000 Worten, das selbst erweitert oder für eine andere Sprache erstellt werden kann.
Regelmäßig, aber nicht mehr mäßig wird **»SysInfo«** von Nic Wilson "upgedatet". Fast entsteht der Eindruck, als würde jede neue Version zuerst zu Fred Fish geschickt und dann erst getestet.
»TplEdit« ist ein Templete-Editor für die »Gadtools.library«, der allein ablauffähigen Code erzeugt, dieser kann dann direkt compiliert werden.
Wiederum von Jonathan Potter stammt **»WorldTime«**, ein Uhrprogramm mit einem besonderen Feature: Auf Wunsch öffnet sich ein größeres Window, das die aktuelle Zeit in 84 großen Städten der Welt anzeigt. Wenn die eigene nicht dabei ist: Das Programm ist da frei konfigurierbar.

FISH 584

584 – Auch wenn der Name anderes vermuten läßt, ist **»DataPlot«** von Stefan Zeiger kein echter Funktionsplotter. Statt dessen wird aus den eingegebenen Parametern ein Datenfile erzeugt, das dann in

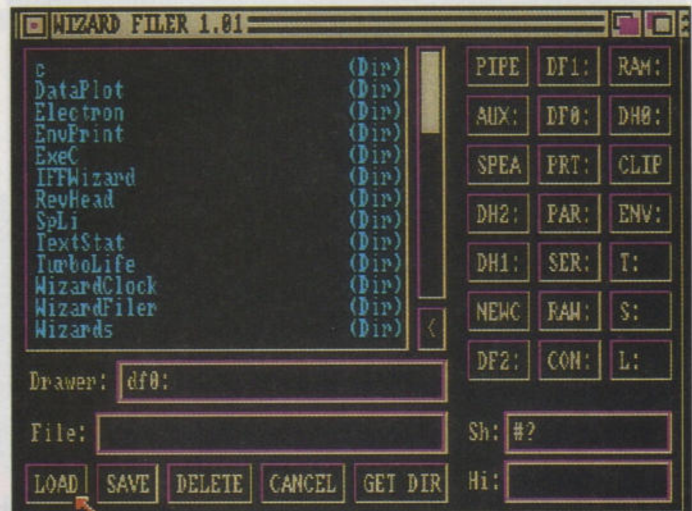
»MultiPlot« (von Fish 467) eingelesen werden kann, wo zur Manipulation weit mehr Möglichkeiten zur Verfügung stehen, als Funktionsplotter normalerweise bieten.
Vom gleichen Autor stammt noch mehr auf dieser Disk. So ist beispielsweise **»Electron«** vertreten, eine Simulation des im März 1990 in Spektrum der Wissenschaft beschriebenen Automaten »Electron-World«.
»EnvPrint« hilft beim Bedrucken von Briefumschlägen. Es muß nur eine Datenbank geladen oder eingegeben werden, und die Umschläge werden automatisch beschriftet.
Der **»IFFWizard«** ist in der Lage, alle Chunks einer IFF-Datei anzuzeigen, wobei er gleichzeitig eine Kurzbeschreibung zum Hunk sowie seine Länge ausgibt. **»TextStat«** ist ein erweitertes "WordCount"-Programm, ähnlich dem »wc« aus dem SAS/C-Paket, jedoch mit der zusätzlichen Möglichkeit, auch die Häufigkeiten einzelner Buchstaben beziehungsweise Zeichen auszugeben. **»TurboLife«** simuliert das bekannte "Life" in einer annehmbaren Geschwindigkeit. **»WizardClock«** ist ein weiteres in der schier endlosen Menge der Uhrprogramme, während der **»WizardFiler«**, der sich an Anders Bjerins »FileWindow« von Fish 337 orientiert, ein komfortabler Datei-Requester für den Einbau in eigene Programme ist.
Die restlichen drei Programme dieser Diskette stammen von Carsten Raufuß. **»ExeC«** ist ein kleines Utility, das es erlaubt, CLI/Shell-Befehle von der Workbench aus zu starten.
»RevHead« hilft bei der Erstellung von programmunabhängigen Info-Files zu Sourcecodes oder Executables. **»SpLi«** generiert hübsch anzusehende sphärische Lissajous-Figuren, die auf einem in Spektrum der Wissenschaft veröffentlichten Algorithmus beruhen.

FISH 585

#585 – Ein kompletter Raytracer namens **»RayDance«** findet sich auf dieser Fish-Disk, allerdings nur in einer Demo-Version, bei der die maximale Anzahl der Objekte jedoch begrenzt ist und beim Berechnen eines Bildes alle Viertelstunde einmal ein Requester beantwortet werden muß, bevor es mit der Berechnung weitergeht. Ansonsten ist das Programm jedoch voll funktionsfähig und wird sowohl in einer Version für normale Amigas als auch in einer Version mit Inline-FPU-Support geliefert.
586 – Christian Friedel zeichnet für **»AzMake«** verantwortlich, eine integrierte Entwicklungsumgebung für Aztec C, die das Compilieren, Assemblieren, Linken, Ausdrucken und mehr direkt per Klick auf ein Gadget erlaubt.

FISH 586

»BlackHole« von Parsec Soft Systems installiert ein Application-Icon auf der Workbench (2.0), das Dateien, die auf ihm abgelegt werden, nach einer Sicherheitsabfrage löscht. Anders als beim Trashcan der Workbench sind die Dateien danach aber tatsächlich gelöscht.
Von Bruce Thomson stammt das Sculpt-4D-Tool **»Contour 4D«**. Mit ihm lassen sich zwei IFF-Bilder zu einem Sculpt-4D-Objekt konvertieren, bei dem die Farbe eines Pixels im ersten Bild die Farbe (und gegebenenfalls Textur) des Objektes an dieser Stelle angibt, die des entsprechenden Pixels im zweiten Bild die Höhe an dieser Stelle.
Stefan Berendes schrieb **»LSlabel«**, ein Utility zum Ausdrucken von Etiketten. Besonderes Feature dieses Programms ist die nahezu völlig freie Konfigurierbarkeit, was das Aussehen der erzeugten Etiketten und die verwendeten Druckercodes angeht.
Das im PD-Pascal PCQ geschriebene **»Opus8«** stammt von John A. Safranek und wandelt Macintosh-8-Bit-Sounds ins IFF-8SVX-Format.
Wieder einmal eine Hangman-Variante ist in Form von Joe Rattz **»TheGallows«** dabei. Eine Datenbasis von über 3000 Worten sorgt dabei dafür, daß das Spiel nicht so schnell langweilig wird, und darüber hinaus kann die Wortliste selbst erweitert werden.
Ein weiteres Sculpt-4D-Tool von Bruce Thomson ist **»Tree 4D«**, das mit relativ vielen, vom Benutzer einstellbaren Parametern automatisch Bäume für Sculpt erzeugt, sogar mit Blättern.



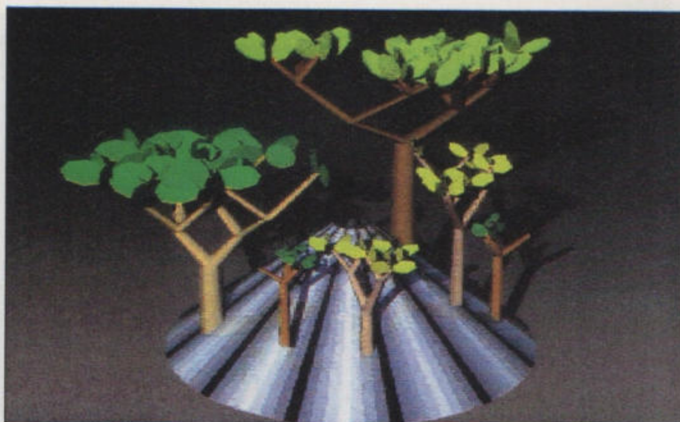
Komfortabler Datei-Requester »WizardFiler«

587 – Björn Reese stellt allen Programmierern seine Runtime-Library **»ConLib«** zur Verfügung. Darin enthalten sind unter anderem Funktionen zum Formatieren und Anzeigen von Text, zur Eingabe von Daten und zur Cursorkontrolle. Die Library ist reentrant (wiederaufruffähiges Programm, ohne den Code aus dem Speicher zu entfernen) und in Assembler-Source verfügbar.

FISH 587

Nach längerer Pause ist auch von Götz Müllers Console-Handler **»GMC«** wieder ein Update in Form der Version 9.11 dabei. Der »GMC« ist einer der komfortabelsten Console-Handler, benötigt aber eine gewisse Zeit der Einarbeitung in die Funktionsvielfalt.
Und **»TMKBP«** von Neil Coito und Michael Cianflone bringt die Kickstart-Umschaltplatine zum Selberbauen. Hinter der Ab-


4D-Effekte mit »Contour 4D«


Bäume in 4D mit »Tree 4D«

kürzung verbirgt sich "The Multi Kickstart Board Project", mit allen Schaltplänen und Layouts für eine Kickstart-Umschaltplatine, die bis zu drei verschiedene Betriebssysteme aufnehmen kann und auch 2.0-tauglich ist.

588 – Matt Dillon hat ein Update zu seiner »fifo.library« beigesteuert. "Fifo" steht für "First In, First Out" und beschreibt ein einer Pipe vergleichbares Device, dessen Vorteil es jedoch ist, auf einer der beiden Seiten statt über die DOS-Funktionen auch über die speziellen Funktionen der »fifo.library« ausgelesen werden zu können.

FISH 588

Auch Olaf Barthel hat wieder einige seiner Programme zur Veröffentlichung eingesandt, die in Deutschland teilweise durch Datennetze schon längst verbreitet sind. Das erste davon ist sein Screenblanker »FracBlank«, nicht irgendein Screenblanker, sondern einer, der Fraktale zeichnet. Die Version 1.8 ist ein Update zur bereits auf Fish 535 veröffentlichten Version 1.4 und enthält nun unter anderem auch einen Farbmodus.

»MandelSquare« ist Olafs auf Turbokarten optimiertes Mandelbrot-Programm. Das nur auf Amigas mit 68020 und höher sowie einem mathematischen Coprozessor lauffähige Programm erlaubt dabei auch die Erstellung von Animationen, die einen Zoom in die Mandelbrot-Menge zeigen und mit jedem ANIM-5-kompatiblen Programm verarbeitet und abgespielt werden können.

FISH 589

589 – **»Term«**, das OS-2.0-Terminal-Programm von Olaf Barthel, ist inzwischen so umfangreich geworden, das trotz Packens großer Teile des Paketes eine ganze Diskette davon gefüllt wird. DFÜ-Anwendern muß man nicht mehr viel zu den Vorzügen von »Term« erzählen, andere werden nicht viel damit anfangen können. Aktuell ist nunmehr jedenfalls die Version 1.9c.

FISH 590

590 – Die letzte der 20 neuen Fish-Disketten beginnt mit **»Crystals«**, einem Programm von David McKinstry. Es handelt sich dabei um eine Computersimulation dreidimensionaler Kristalle, die in Echtzeit gedreht und positioniert werden können, und ist in erster Linie als Anschauungsmaterial zu einem Lehrgang in Kristallographie gedacht, aber auch so nett anzuschauen.

»MICE« steht für "My Image Code Editor" und stammt von Pierre-Louis Mangeard. Es wandelt IFF-Bilder in Include-Files für Assembler oder C um. Vom selben Autor stammt auch **»Equilog«**, eine Umsetzung des bekannten Gesellschaftsspiels "Superhirn".

»Seeker« ist ein Utility zum automatisierten Suchen von Dateien auf einer Festplatte, das mit einer grafischen Benutzeroberfläche versehen wurde. Donald Loyd hat dieses Shareware-Programm entwickelt.

Abgeschlossen wird der Reigen der neuen Fish-Disketten schließlich von Sebastiano Vigna, der mit der Version 2.01 das erste Update zu **»SuperDuper 2.0«** veröffentlicht. Neu sind ein "NoClick"-Utility sowie ein Programm zum Patchen einiger interner Parameter des Trackdisk-Devices, die nunmehr mit im Lieferumfang sind. ❑

Holger Lubitz (vb)

Eine kleine Übersicht der Vertreiber von Public Domain, Free- und Shareware sowie Prüf-vor-Kauf-Programmen (ohne Anspruch auf Vollständigkeit)

APS Electronic, Sonnenborstel 31, 3071 Steimbke, Tel.: 05026/1700
Amiga PD, Pfarrgasse 23, 6670 St. Ingbert, Tel.: 06894/381331
Computerzubehör Hager, Bahnhofstr. 169, 4370 Marl-Sinsen, Tel.: 02365/81629
Delta PD-Service, Schwalbacherstr. 61, 6200 Wiesbaden, Tel.: 0611/379189
Donau-Soft, Postfach 1401, 8858 Neuburg, Tel.: 08431/49798
R. Dombrowski, Postfach 710462, 2000 Hamburg 71, Tel.: 040/6428225
Kaminski, Walter, Geilenkirchener Str. 4, 5110 Alsdorf, Tel.: 02404/22963
Keim, Peter, Vogelsanger Str. 34, 5000 Köln 30, Tel.: 0221/520765
A. Manewaldt, Postfach 129, 6703 Limburgerhof, Tel.: 06236/67300
MVC, Hammer Str. 103, 4730 Ahlen, Tel.: 02382/2503
Nordsoft, Schwenecker & Behnke, Heidbergstr. 75, 2800 Bremen 21, Tel.: 0421/611430
Patrick Pawlowski, Ellerbruch 19, 2177 Wingst, Tel.: 04778/7294
PD-Center, Postfach 3142, 5840 Schwerte 3
PD-Software, Marco Gerlach, Schwalbenweg 14, 1188 Schoenefeld
Pielago Software, Olpener Str. 438, 5000 Köln 91, Tel.: 0221/8903162
Renner's PD-Soft, Fürst-Franz-Josef-Str. 14, 5305 Alfter, Tel.: 02222/61424
R&M, Damaschkestr. 1, 4050 Mönchengladbach 2, Tel.: 02166/850516
RS-Shop, Oberwürzbacherstr. 22, 6676 Mandelbachtal 3, Tel.: 06803/3338
W & L Computer, Okerstr. 46, 1000 Berlin 44, Tel.: 030/6227371

Österreich

M.A.R-Computershop, A-1100 Wien, Weldengasse 41, Tel.: 0222/621535

Schweiz

Swissoft, Miklos Daszkel, Marktgasse 20, CH-2502 Biel, Tel.: 032/225750
Mailsoft Switzerland, Alex Brander, Feldstr. 20, CH-8330 Pfaeffikon ZH, Tel.: (0) 1 950 5611

SHOPPYS TIPS VON A-Z

Einkaufsführer

- **ACD GmbH**, Dammweg 15, 2800 Bremen 1, Tel.: 0421/343131
Activa International, Keyenbergweg 95, 1101 GE Amsterdam Zuidoost, Tel.: 0031/20/6911914
Advanced Systems & Software, Homburger Landstr. 412, 6000 Frankfurt 50, Tel.: 069/5488130
Aeon Verlag & Studio, Frauenhoferstr. 51b, 6450 Hanau 1, Tel.: 06181-23525
A&L AG, Däderiz 61, CH-2540 Grenchen, Schweiz, Tel.: (0041) 065/520311
ALPHA 2000 GmbH, Wasgaustr. 34, 6230 Frankfurt/M.-Höchst, Tel.: 069/300015+16
amigaOberland, Hohenwaldstr. 26, 6374 Steinbach, Tel.: 06171/71846
AmTeK, G. Schneider, Schadewitzstr. 22, 8700 Würzburg, Tel.: 0931/887124
ARTware COM GmbH, Liebigstr. 1, 4500 Osnabrück, Tel.: 0541/259211
AURA, F. Sprenger, Brassertstr. 18, 4300 Essen 1, Tel.: 0201/770826

- **BCT**, Westwall 4, 4270 Dorsten, Tel.: 02362/42991-42925
Bittner, Wilh.-v.-Ketteler-Str. 5, 6707 Schifferstadt, Tel.: 06235/1070
Blue Data GmbH, Heiligenstr. 30-32 4010 Hilden, Tel.: 02103/69019
Bomico, Am Südpark 12, 6092 Kelsterbach, Tel.: 06107/62067
bsc Büroautomation AG, Lerchenstr. 5, 8000 München 50, Tel.: 089/357130-0

- **CAS Computer Animation Studio**, Reihstr. 67, 5100 Aachen, Tel.: 0241/31991
C-Data, Hohenwarter Str. 6, 8068 Pfaffenhofen, Tel.: 08441/6145
ComData, Schiffgraben 19, 3000 Hannover 1, Tel.: 0511/326736
Commodore Büromaschinen GmbH, Lyoner Str. 38, 6000 Frankfurt 1, Tel.: 069/6638-0
CompuStore, Fritz-Reuter-Str. 6, 6000 Frankfurt 1, Tel.: 069/567399
Compy Shop, Missundestr. 48, 4600 Dortmund, Tel.: 0231/818023
CRP Koruk, Fritz-Arnold-Str. 23, 7750 Konstanz, Tel.: 07531/56265
CSV Riegert GmbH, Gärtnerstr. 4, 7320 Göppingen, Tel.: 07161/13591

- **Delta Konzept GmbH**, Bösinghovener Str. 98, 4005 Meerbusch, Tel.: 02159/81007
Demonware, Strahlenburger Str. 125, 6050 Offenbach, Tel.: 069/8004703
DMV-Verlag, Fuldaer Str. 6, 3440 Eschwege, Tel.: 05651/809-0
Dreyer Automationssysteme, An der Bornau 3, 3060 Stadthagen, Tel.: 05721/72360
DTM Computersysteme, Dreiherrenstein 6a, 6200 Wiesbaden-Auringen, Tel.: 06127/4065

- **EchtZeit**, Friedensallee 14-16, 2000 Hamburg 50, Tel.: 040/3905811
ECS Vertriebs GmbH, (Leisuresoft), Rosenheimer Str. 92a, 8000 München 80
Elmar Elsner, Computer & Peripherie, Fröbelstr. 57, 5000 Köln 71, Tel.: 0221/5906242
ETS, Tel.: 08124/7677
European Software Distributors, 5223 Nümbrecht-Oberbech 1, Tel.: 02262/5898

- **Fischer Hard- & Software**, Schierholzstr. 33, 3000 Hannover 51, Tel.: 0511/572358, 575087
FSE Computer-Handels GmbH, Schmiedstr. 11, 6750 Kaiserslautern, Tel.: 0631/6441

- **GTI GmbH**, Zimmersmühlenweg 73, 6370 Oberursel, Tel.: 06171/73048-9

- **Hirsch&Wolf OHG**, Computer & CD, Mittelstr. 33, 5450 Neuwied, Tel.: 02631/24485
HK Computer, Bonner Str. 37, 5000 Köln 1, Tel.: 0221/311606
HS&Y OHG, Classen-Kappelmann-Str. 24, 5000 Köln 41, Tel.: 0221-404078

- **Jin Tech**, Ampertal 8, 8049 Unterbrugg/Fahrenzhausen, Tel.: 08133/2044-5
Joystick, Lübecker Str. 82, 2000 Hamburg 76, Tel.: 040/25(1)4592

- **Kingsoft**, Grüner Weg 29, 5100 Aachen, Tel.: 0241/152051

- **Leisuresoft**, Robert-Bosch-Str. 1, 4703 Bönen, Tel.: 02383/690

- **Mac Soft**, Wilhelmstr. 33, 4600 Dortmund 1, Tel.: 0231/161817
Maxon Computer GmbH, Schwalbacher Str. 52, 6236 Eschborn, Tel.: 06196/481811
Memphis Computer Products, Gartenstr. 11, 6365 Rodheim v.d. Höhe, Tel.: 06007/7789
Merkens EDV, Fuchstanzstr. 6a, 6231 Schwalbach, Tel.: 06196/3026
MVC, Hammer Str. 103, 4730 Ahlen, Tel.: 02382/2503

- **NEC Deutschland GmbH**, Klausenbergerstr. 4, 8000 München 80, Tel.: 089/93006-0

- **Hardware Design Neuroth**, PF 525, 4250 Bottrop, Tel.: 02041/20424

- **OMEGA Datentechnik**, Junkerstr. 2, 2900 Oldenburg, Tel.: 0441/82257

- **PBC**, Peter Biet, Dietershausener Str. 28, 6409 Dipperz, Tel.: 06657/8606
Pictures of Oz-Medienproduktion, Hochbrücker Mühle 2-3, 5100 Aachen, Tel.: 0241-151109
- **Pro-Computer**, Schläger&Eisen-Str. 46, 4352 Herten, Tel.: 02366/55176
ProFont, Mailingerstr. 20, 8000 München 19, Tel.: 089/1231181
Prowort, PF 1316, 8045 Ismaning, Tel.: 089/966270
protar Elektronik GmbH, Alt Moabit 91d, 1000 Berlin 21, Tel.: 030/3912002
Pulsar, Erlanger Str. 8-10, 5000 Köln 91, Tel.: 0221/873359

- **reLINE**, Königstr. 55, 3000 Hannover 1, Tel.: 0511/315834/344521
Rushware, Hochweg 128-132, 4044 Kaarst 2, Tel.: 02101/6070

- **Serafin Software**, Messerschmidtg. 40/1, A-1180 Wien, Österreich, Tel.: (0043) 01/4700525
Softgold GmbH, (Rainbow Arts), Daimlerstr. 10, 4044 Kaarst 2, Tel.: 02101/66020
Software 2000, Lübecker Str. 10, 2320 Plön, Tel.: 04522/1379
Supra (Deutschland) GmbH, Rodderweg 8, 5040 Brühl, Tel.: 02232/22002

- **Thumm**, Cheruskerstr. 21, 7036 Schönaich, Tel.: 07031/52664
TKR, Projensdorferstr. 14, 2300 Kiel 1, Tel.: 0431/337881
Tronic-Verlag, Fuldaer Str. 6, 3440 Eschwege, Tel.: 05651/929-0

- **UGA Software**, P.O. Box 881, 3700 AW Zeist, Niederlande, Tel.: (0)3404/53531
United Software, PF 2351, 4835 Rietberg, Tel.: 05244/40820

- **vortex**, Falterstr. 51-53, 7101 Flein, Tel.: 07131/5972-0

- **WALLASCH & WITTE GmbH**, PF 1025, 8000 München 81, Tel.: 089/938224
Wiegand Video Datensysteme, Palmersdorfer Hof 11-19, 5040 Brühl, Tel.: 02232/45028
Woerlein GmbH, PF 4, Gewerbepark Schwadermühle, 8501 Cadolzburg, Tel.: 09103/8294, 8552, 905
Word Perfect GmbH, Frankfurter Str. 21-25, 6236 Eschborn, Tel.: 06196/904-01

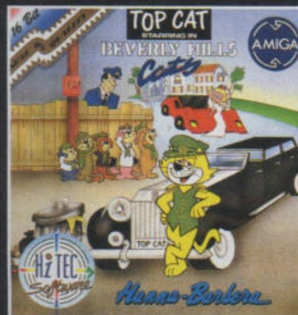
Inserenten

Ein Teil der Auflage enthält Beilagen der Firma Westfalia Technica.
Die ABO Auflage enthält Beilagen der Firma Westfalia Technica.

SPIELESHOW

AMIGA DOS

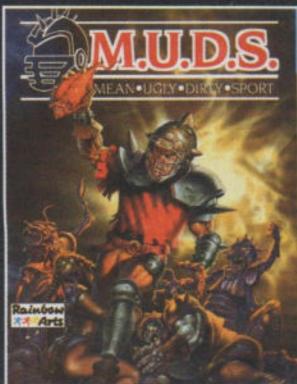
Gelingt es Top Cat, die spannenden Erbstreitereien um Madame Van Der Gelt trotz aller Escapaden zu schlichten?



20,– DM*

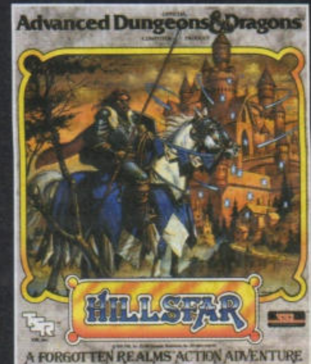
Das ultimative Sportspiel macht Sie zum Manager, der das Team wieder aufrichtet und zum Sieg führt, denn die Moral ist durch Betrug, Unfairneß, Kredithaie und Prügeleien am Boden. Mit 16 Städten und 16 Charakteren; auch für 2 Spieler.

69,– DM*



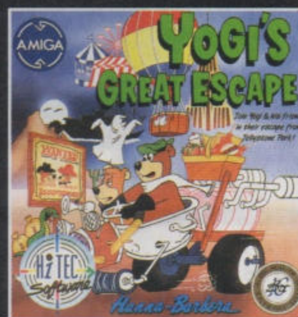
Die Stadt in den magischen "Forgotten Realms", einer Spielwelt reich an Rätseln und Herausforderungen, die unglaubliche Fantasiewelt, die die besten Rollenspiele und atemberaubendsten Abenteuerspiele kombiniert.

49,– DM*



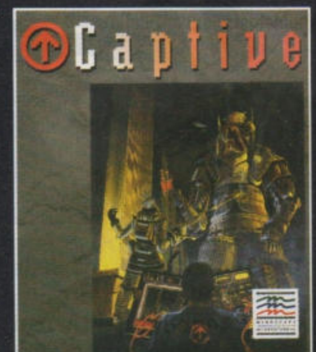
Yogi nimmt wieder sein alljährliches Hobby auf, nichtsahnenden Touristen die Picknickkörbe zu stibitzen.

20,– DM*



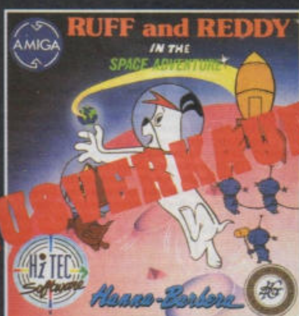
200 Jahre lang versuchen Sie, unschuldig, aus dem elektronischen Space-Prison mit Hilfe eines kleinen Computers zu entkommen. Dann beginnt "the Quest for freedom", als seltsame Droiden Hilfe anbieten.

49,– DM*



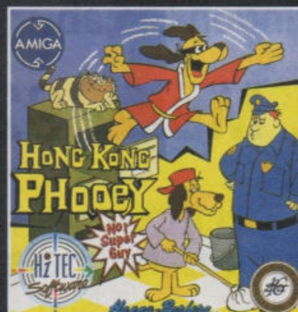
Professor Flipnoodles Taschenrakete verirrt sich auf den Planeten der Lilli-Punies.

20,– DM*



Baron von Bankjob ist aus dem Gefängnis entflohen. Nur Hong Kong Phooey scheint die Situation retten zu können.

20,– DM*

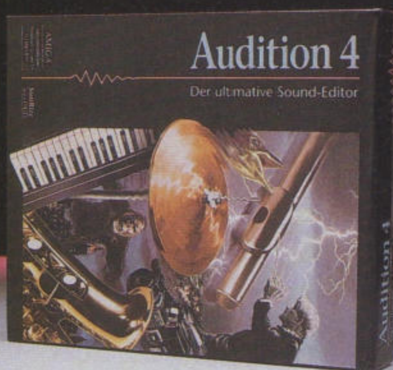


* Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · W-3440 Eschwege

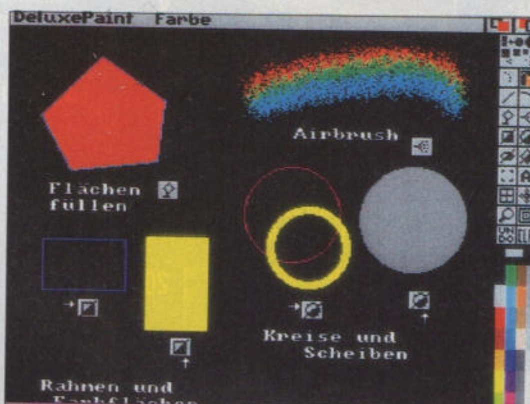
DMV Daten- und Medienverlag

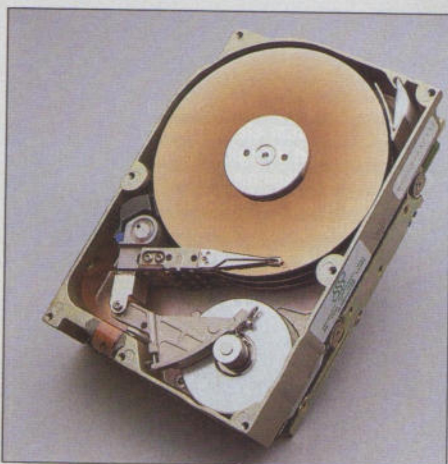
Vorschau



»Audition 4« – ein Nachfolger für »AudioMaster«? Dieser neue Sample-Editor soll das Erstellen von digitalisierten Audio-Aufnahmen zum Kinderspiel machen. Wir haben ihn für Sie geprüft und dem »AudioMaster IV« gegenübergestellt. Welcher der beiden als Sieger hervorgeht, erfahren Sie in der nächsten AMIGA DOS.

Kurse und Workshops, Programme und Programmiertips – in der nächsten Ausgabe gibt es wieder eine bunte Fülle an allem, was dem Amiganer zu Hause fehlt. Es geht weiter in unserem Kurs über »DPaint IV«, die Shell wird in den Betriebssystem-Versionen 1.3 und 2.0 weiter beleuchtet, und bei den Tips & Tricks gibt's wieder viel zu lernen.





Das Titelthema der nächsten Ausgabe heißt "Aufrüsten". Keine Angst, es geht hier nicht um Waffen, sondern um die Peripherie für den Amiga.

Impressum

Herausgeber
Christian Widuch

Redaktionsleitung
Matthias Bloß

Chefredakteur
Bernd Zimmermann (verantw.)

Leitender Redakteur
Jürgen Borngießer (jb)

Redaktion
Thomas Baum (tb), Vera Brinkmann (vb), Garry Glendown (gg), Heinrich Stiller (hs)

Freie Autoren dieser Ausgabe
Heike Albrecht, Ute Bahn, Arno Eigenwillig, Frank Gärtner, Michael-Wolfgang Hohmann, Dr. Peter Kittel, Holger Lubitz, Sandro Paolini, Siegfried Rings, Hanns Holger Rutz, Hartmut Schumacher, Gerald Seidler, Christoph Teuber, Holger Wallmann, Dusan Zivadinovic

Redaktionsassistenz
Susanne Reckelkamm (re), Carmen Strube

Produktionsleitung
Uwe Siebert

Fotosatz
Marcus Geppert, Andrea Kloss

Layout
Patricia Reifenhausen (Ltg.), Andrea Kloss

Montage/Reprografie
Margarete Schenk (Ltg.), Helmut Skoupy (Stellv.), Susanne Eckhardt, Manuela Eska, Andrea Herschelmann

Fotografie
Klaus Jatho, Claudia Schlott

Art Director
Mohamed Hawa

Eigenwerbung
Uwe Kalischinski, Patricia Reifenhausen, Norbert Risch

Anzeigenleitung
Wolfgang Brill

Anzeigenverkauf
DMV-Verlagsbüro München, Zaunkönigweg 2c, 8000 München 82
Telefon: (089) 439 10 87, Telefax: (089) 439 10 80
Leitung: Britta Fiebig
Anzeigenverkauf: Peter Schätzle, Hannelore David, Ilona A. Sehm

Coordinating Manager Asia
Ralph Streng

Coordinating Manager USA
Tobias J. Hoffmann

Repräsentant im Ausland
Great Britain
Huson European Media
Gerry Rhoades Brown
10-11 The Green Business Centre
The Causeway
Staines
Middlesex
TW18 3AL
Telephone No.: 0044-784 469900, Fax No.: 0044-784 469996

Anzeigenverwaltung und Disposition
Andrea Giese, Beate Kranz

Druckunterlagen-Disposition
Christina Wabra (05651) 809-372

Anzeigenpreise
Es gilt die Anzeigenpreisliste Nr. 3 vom 01.11.1991

Anschrift Verlag/Redaktion
DMV Daten- und Medienverlag, Widuch GmbH & Co. KG
Fuldaer Straße 6, 3440 Eschwege,
Telefon (05651) 809-0, Telefax (05651) 809-333

Vertrieb
Verlagsunion Erich Pabel-Arthur Moewig KG (VPM),
Friedrich-Bergius-Straße 20, 6200 Wiesbaden

Druck
Druckerei Jungfer, 3420 Herzberg

Bezugspreise
»AMIGA DOS« erscheint monatlich.
Einzelpreis DM 6,50/sfr 6,50, öS 52,–, Lit 6000, Ptas. 520

Abonnementpreise
Die Preise verstehen sich grundsätzlich einschließlich Porto und Verpackung.

Inland:
12 Ausgaben: DM 70,–
6 Ausgaben: DM 35,–
Europäisches Ausland:
12 Ausgaben: DM 100,–
6 Ausgaben: DM 50,–
Außereuropäisches Ausland:
12 Ausgaben: DM 120,–
6 Ausgaben: DM 60,–

Bankverbindungen:
Postscheck Frankfurt/M: Kto-Nr.: 23043-608
Raiffeisenbank Eschwege: BLZ: 522 603 85, Kto-Nr.: 245 7008

Die Abonnementbestellung kann innerhalb einer Woche nach Auftrag beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich widerrufen werden. Zur Wahrung der Frist reicht der Poststempel. Das Abonnement verlängert sich automatisch um 6 bzw. 12 Monate, wenn es nicht mindestens 6 Wochen vor Ablauf beim Verlag schriftlich gekündigt wird.



Mitglied der Informationsgemeinschaft zur Feststellung der Verbreitung von Werbeträgern e.V. (IVW), Bad Godesberg.
ISSN 0937-2717

REFLECTIONS 2.0

DIE NEUE GENERATION DES RAYTRACING

Der erfolgreichste deutsche Raytracer hat einen würdigen Nachfolger gefunden: **Reflections 2.0!**

Neueste Berechnungsmethoden und ausgeklügelte Modellierfunktionen lassen fotorealistische Bilder auf Ihrem Amiga entstehen, die den Ergebnissen großer Workstations kaum nachstehen!

Vier Editoren bieten Ihnen die Möglichkeit, jede nur erdenkliche Form mit wenig Aufwand zu erstellen. Diverse Tools erlauben Ihnen die mühelose Generierung von 3D-Fonts oder auch von Schläuchen, Röhren und Spiralen. Darüber hinaus machen einzigartige Leckerbissen wie eine Nebelfunktion, Boole'sche Operationen oder das Bumpmapping **Reflections 2.0** zu einem der umfangreichsten und professionellsten Raytracer für den Amiga.

Reflections 2.0 DM 349,–*

UPDATE Senden Sie bitte Ihre Reflections-Original-Disketten zusammen mit einem Verrechnungsscheck an M&T Software Partner International GmbH, Update-Service, Hans-Pinsel-Straße 9b, 8013 Haar

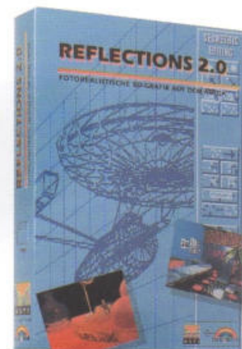
Reflections 2.0 Update (Best.-Nr. 54149A) DM 249,–*

Jede Vorgängerversion von Reflections kann upgedatet werden!

Systemanforderungen:
Alle Amiga mit mind. 1 MB, Kickstart 1.3, OS 2.0

Mailbox-Service – 7 Tage die Woche, 24 Stunden am Tag! Mailbox-Nr.: (0 89) 46 15 15 Fordern Sie unseren Antrag an!

* Unverbindliche Preisempfehlung





EIN UNTERNEHMEN VON

